

# **IDC RE-ENGINEERING REPORT**

SAND2016-**XXYYZZ**

Unlimited Release

June, 2016

## **IDC Re-Engineering Phase 2 Iteration E2 Use Case Realizations**

Version 1.0

J. Mark Harris, John F. Burns, Benjamin Hamlet, Randall Lober, James Vickers

Prepared by  
Sandia National Laboratories  
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.



SAND2016-**XXYYZZ**  
Unlimited Release  
June, 2016

# **IDC Re-Engineering Phase 2 Iteration E2 Use Case Realizations**

J. Mark Harris, John F. Burns  
Dynamic Monitoring Software

Benjamin Hamlet, Randall Lober, James Vickers  
Ground System Development

Sandia National Laboratories  
P.O. Box 5800  
Albuquerque, New Mexico 87185-MS0401

## **Abstract**

This document contains 4 use case realizations generated from the model contained in Rational Software Architect.

## REVISIONS

Version	Date	Author/Team	Revision Description	Authorized by
1.0	06/01/2016	SNL IDC Re-Engineering Team	Initial Release for E2	M. Harris

## TABLE OF CONTENTS

Use Case Hierarchy.....	6
UCR-01.04 System Acquires Meteorological Data.....	9
UCR-02.06 System Builds Event Using Signal Detections.....	23
UCR-03.03 Scans Waveforms and Unassociated Detections.....	57
UCR-14.05 Performs Expert Technical Analysis.....	81

# Use Case Hierarchy

The IDC Use Case Hierarchy is shown here. The use cases highlighted in yellow are the use case realizations that appear in this document.

## **1 System Acquires Data**

- 1.1 System Receives Station Data
- 1.2 System Receives Bulletin Data
- 1.3 System Automatically Distributes Data
- 1.4 System Acquires Meteorological Data
- 1.5 System Synchronizes Acquired Station Data
- 1.6 System Synchronizes Processing Results

## **2 System Detects Event**

- 2.1 System Determines Waveform Data Quality
- 2.2 System Enhances Signals
- 2.3 System Detects Events using Waveform Correlation
- 2.4 System Detects Signals
- 2.5 System Measures Signal Features

## **2.6 System Builds Events using Signal Detections**

- 2.7 System Resolves Event Conflicts
- 2.8 System Refines Event Location
- 2.9 System Refines Event Magnitude
- 2.10 System Evaluates Moment Tensor
- 2.11 System Finds Similar Events
- 2.12 System Predicts Signal Features

## **3 Analyzes Events**

- 3.1 Selects Data for Analysis
- 3.2 Refines Event
  - 3.2.1 Determines Waveform Data Quality
  - 3.2.2 Enhances Signals
  - 3.2.3 Detects Signals
  - 3.2.4 Measures Signal Features
  - 3.2.5 Refines Event Location
  - 3.2.6 Refines Event Magnitude
  - 3.2.7 Evaluates Moment Tensor
  - 3.2.8 Compares Events

## **3.3 Scans Waveforms and Unassociated Detections**

- 3.4 Builds Event
- 3.5 Marks Processing Stage Complete

## **4 N/A**

## **5 Provides Data to Customers**

- 5.1 Requests System Data

5.2 Approves Events for External Release

5.3 Views System Results

## **6 Configures System**

6.1 Controls Data Acquisition

6.2 Configures Station Usage

6.3 Defines Processing Sequence

6.4 Configures Data Acquisition

6.5 Configures Processing Components

6.6 Configures System Messages

6.7 Views System Configuration History

6.8 Configures Analysis Interfaces

6.9 Configures System Permissions

## **7 Monitors Performance**

7.1 Analyzes Mission Performance

7.2 Monitors System Performance

7.3 Monitors Station State-of-Health

7.4 System Monitors Mission Performance

7.5 Monitors Mission Processing

## **8 Supports Operations**

8.1 Accesses the System

8.2 Controls the System

8.3 Exports Data

8.4 Imports Data

8.5 Views Event History

8.6 Maintains Operations Log

8.7 Provides Analyst Feedback

8.8 Views Analyst Feedback

8.9 Views Analyst Performance Metrics

8.10 Views Security Status

8.11 Views Messages

## **9 Tests System**

9.1 Performs Software Component Testing

9.2 Creates Test Data Set

9.3 Replays Test Data Set

9.4 Replays Analyst Actions

## **10 Maintains System**

10.1 Performs System Backups

10.2 Performs System Restores

10.3 Installs Software Update

10.4 System Monitors Security

## **11 Performs Research**

11.1 Analyzes Special Events

- 11.2 Develops New Algorithms and Models
- 11.3 Determines Optimal Processing Component Configuration
- 11.4 Performs Multiple Event Location

## **12 Performs Training**

- 12.1 Configures Data for Training Subsystem
- 12.2 Trains Analysts

## **13 Operates Standalone Subsystem**

- 13.1 Conducts Site Survey
- 13.2 Performs Standalone Analysis

## **14 IDC Unique**

- 14.1 System Assesses Event Consistency
- 14.2 Assesses Event Consistency
- 14.3 System Screens Event
- 14.4 Controls Monitoring Stations
- 14.5 Performs Expert Technical Analysis



# IDC Use Case Realization Report

## **UCR-01.04 System Acquires Meteorological Data**

### **Use Case Description**

This architecturally significant use case describes how the System acquires meteorological data to build atmospheric models used in automatic and interactive processing of infrasound data. The System requests the latest available high-resolution global meteorological data from external data centers and puts it into the correct formats for generation of infrasound propagation models. The system moves the meteorological data from Data Acquisition Partition to the Data Processing Partition and stores the meteorological data. The System builds a new atmospheric model based on the meteorological data.

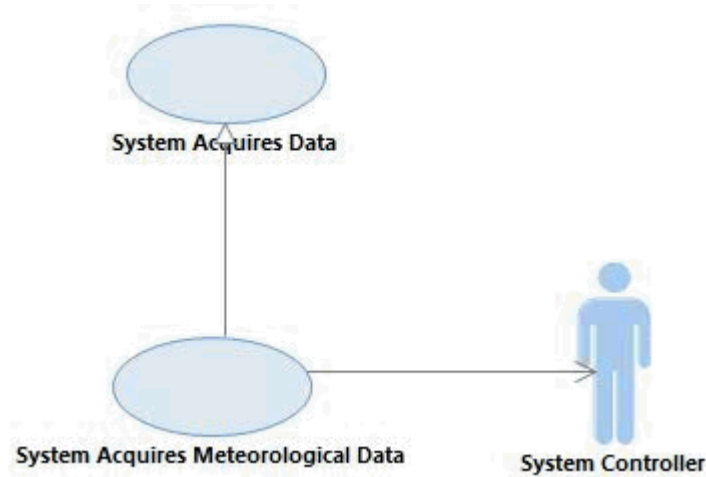
This use case is architecturally significant because it describes acquiring meteorological data from various sources and creating dynamic atmospheric transmission model to support the prediction of infrasonic signal detections.

### **Architecture Description**

The Meteorological Data Acquisition Control is the control class that coordinates the acquisition of meteorological data into the system. The Meteorological Data Acquisition Control reads an instance of Meteorological Data Acquisition Control Configuration from the OSD, and requests new meteorological data at the configured time intervals. If configured, Meteorological Data Acquisition Control saves the data received from External Data Centers to the OSD before conversion. The Meteorological Data Acquisition Control passes the data replies received from the External Data Centers to the Meteorological Data Format Converter utility class, which merges the raw data sets and creates an instance of the System Format Meteorological Data entity class.

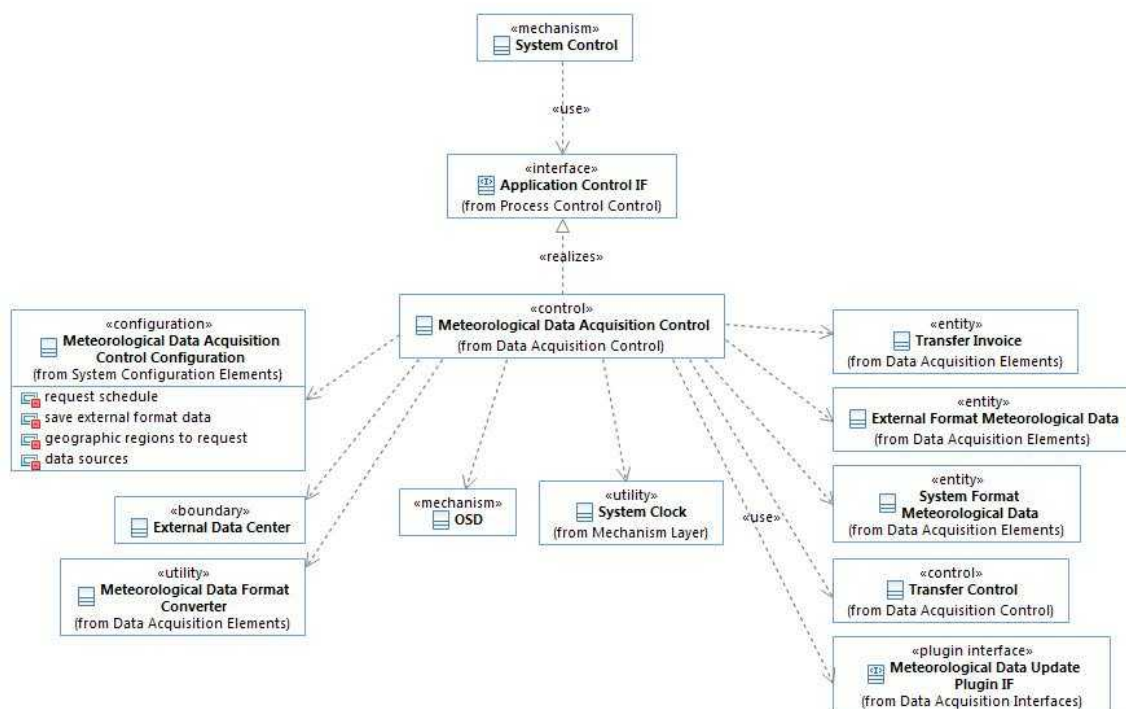
The Meteorological Data Acquisition Control then makes a request to Transfer Control that the System Format Meteorological Data be transferred from the Data Acquisition Partition to the Data Processing Partition. The Meteorological Data Acquisition Control running on the Data Processing Partition processes the transferred meteorological data. The Meteorological Data Acquisition Control passes the System Format Meteorological Data to the Atmospheric Model Plugin through the Meteorological Data Update Plugin IF Interface. The Atmospheric Model Plugin uses the System Format Meteorological Data it receives to build an atmospheric model with possible additional processing such as gravity wave corrections. The Atmospheric Model Plugin provides access to its model, which can be used for infrasound signal propagation predictions (see ‘System Predicts Signal Features’ UCR), through the Earth Model Plugin interface.

## Use Case Diagram



## Class Diagrams

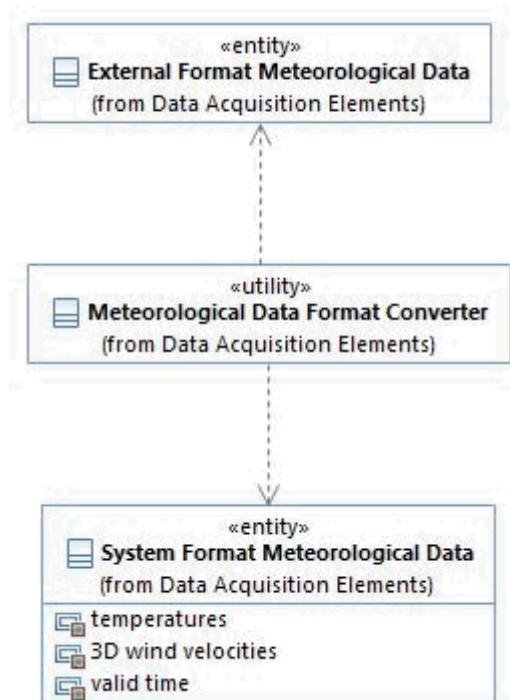
### Classes - Meteorological Data Acquisition Control



This class diagram depicts the Meteorological Data Acquisition Control class and classes it interacts with to request, acquire, and process External Format Meteorological Data from an External Data Center. Meteorological Data Acquisition Control requests updated data from an External Data Center when stimulated by the System Clock, uses Meteorological Data Format Converter to convert the received data into a System Format Meteorological Data, and then uses

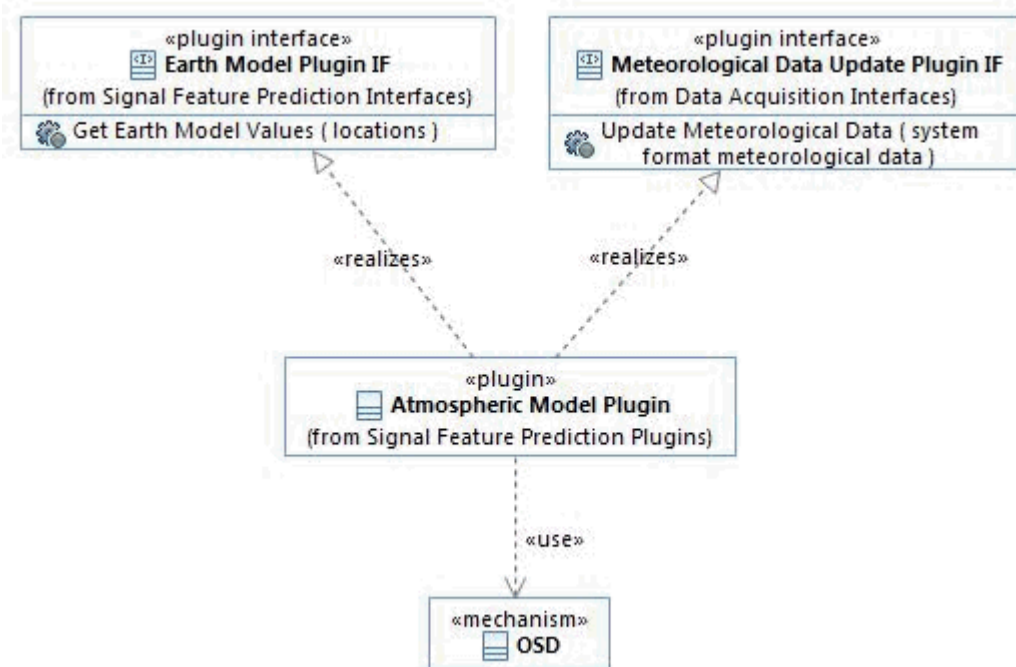
Transfer Control to transfer the System Format Meteorological Data from the Data Acquisition Partition to the Data Processing Partition.

***Classes - Meteorological Data Formats and Format Conversion***



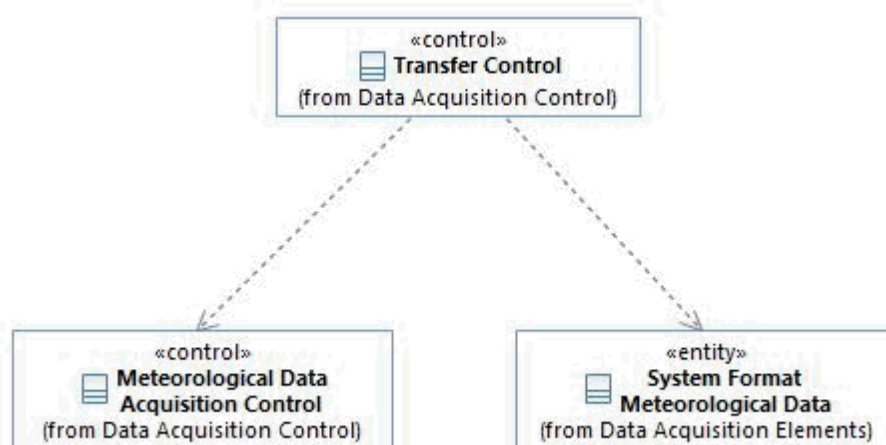
This diagram shows the classes that represent meteorological data in differing external and internal System formats and the Meteorological Data Format Converter utility that converts from the External Meteorological Data format to the System Format Meteorological Data. There are two different formats of Meteorological Data because the format the System receives from an External Data Center may differ from the internal representation. Meteorological Data Acquisition Control uses this utility to convert formats when it acquires data in the External Meteorological Data format.

### Classes - Atmospheric Model Plugin



This diagram shows the Atmospheric Model Plugin and the two plugin interfaces it realizes. Atmospheric Model Plugin provides a kind of earth model by realizing the Earth Model Plugin IF interface and updates that model via the Meteorological Data Update Plugin IF interface.

### Classes - Transfer Control



This diagram shows the Transfer Control class and other classes it uses when transferring System Format Meteorological Data from the Data Acquisition Partition to the Data Processing Partition. Meteorological Data Acquisition Control automatically requests a data transfer after acquiring meteorological data and the System Controller manually requests data transfers using the Request Transfer Display. Transfer Control on the Data Processing Partition stores the Transfer Invoice and the transferred meteorological data in the OSD and uses Transfer Audit Utility to

verify the data transferred correctly. Transfer control can be configured to retransmit data that fails to transfer from the Data Acquisition to the Data Processing Partition, or it can just notify the System Controller of such an occurrence.

## **Class Descriptions**

### **<<boundary>> *External Data Center***

Represents an external data center.

### **<<configuration>> *Meteorological Data Acquisition Control Configuration***

Represents configuration information such as whether to save external format data, meteorological data sources to request from, how often to request data, what data to request (geographic regions), and the sources from which to make requests.

### **<<control>> *Meteorological Data Acquisition Control***

Responsible for coordinating the acquisition and storage (via the OSD) of meteorological data in the System, and for updating an atmospheric model via the Meteorological Data Update Plugin IF Interface.

### **<<control>> *Transfer Control***

Responsible for transfer of data (e.g. Raw Station Data Frames, meteorological data, bulletins, etc.) from the Data Acquisition Partition to the Data Processing Partition. A separate instance runs on both partitions to coordinate the transfer.

### **<<entity>> *External Format Meteorological Data***

Represents meteorological data in the original format provided by an External Data Center (e.g. GRIB2 format). This data includes temperatures, 3D wind velocities, and a time for which the data is considered valid. Meteorological Data Format Converter converts this data into a System Format Meteorological Data object.

### **<<entity>> *System Format Meteorological Data***

Represents meteorological data in the internal format stored on the System and used by Meteorological Data Update Plugin IF to update an Atmospheric Model Plugin.

### **<<entity>> *Transfer Invoice***

A list of transferred data (e.g. Raw Station Data Frames, System Format Meteorological Data) from the Data Acquisition Partition to the Data Processing Partition, and information about previous transfers.

### **<<interface>> *Application Control IF***

Defines the interface implemented by all <<control>> classes in the system that are controlled by System Control.

### **<<mechanism>> *OSD***

Represents the Object Storage and Distribution mechanism for storing and distributing data

objects internally within the system.

**<<plugin interface>> *Earth Model Plugin IF***

Standard interface for all Earth Model plugins. All Earth Model plugins in the system realize this interface.

**<<plugin interface>> *Meteorological Data Update Plugin IF***

This interface allows a class to take an update to the meteorological data it uses. If necessary, the plugin is responsible for storing updated information (e.g. an updated atmospheric model) in the OSD.

**<<plugin>> *Atmospheric Model Plugin***

Provides atmospheric data by implementing the Earth Model Plugin interface. This atmospheric data includes, but is not limited to 3D wind velocity and temperature at altitudes ranging from 0km to 140 km above sea level. The model this plugin provides may also contain gravity wave corrections made to temperature predictions and wind velocities. The Atmospheric Model Plugin may periodically update its atmospheric model when it receives new meteorological data via the Meteorological Data Update Plugin IF interface. Atmospheric Model Plugin is responsible for storing the update model in the OSD.

**<<utility>> *Meteorological Data Format Converter***

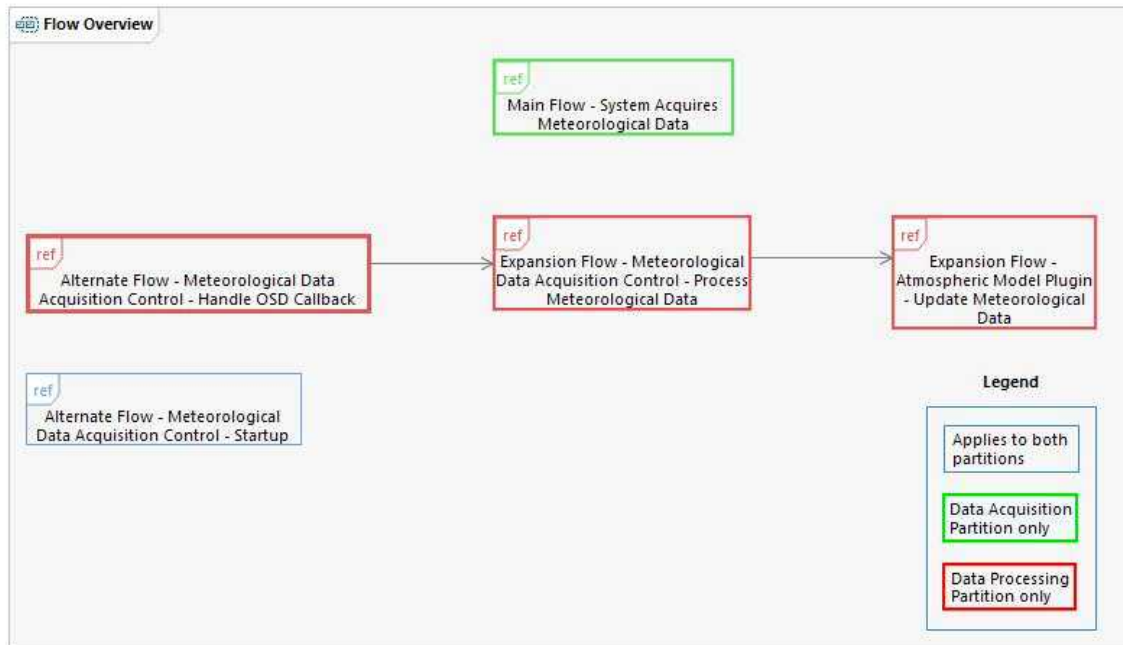
Responsible for converting data received from External Data Centers in the External Meteorological Data format into the System Meteorological Data Format. May merge meteorological data from multiple data centers (e.g. data for different altitude ranges; data for different geographic regions) when creating the System Meteorological Data Format object.

**<<utility>> *System Clock***

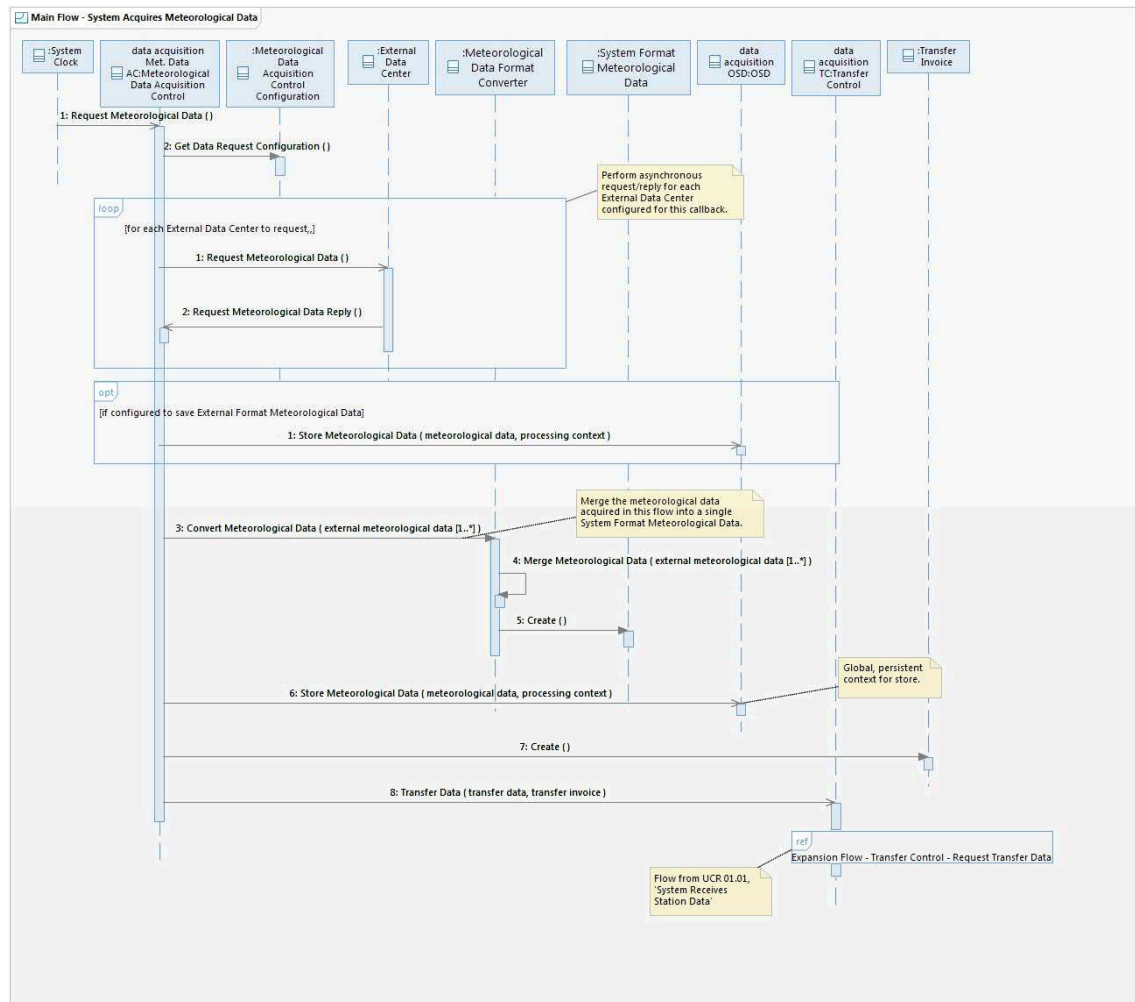
Represents the mechanism to schedule, reschedule, and cancel callbacks.

# Sequence Diagrams

## Flow Overview



## Main Flow – System Acquires Meteorological Data



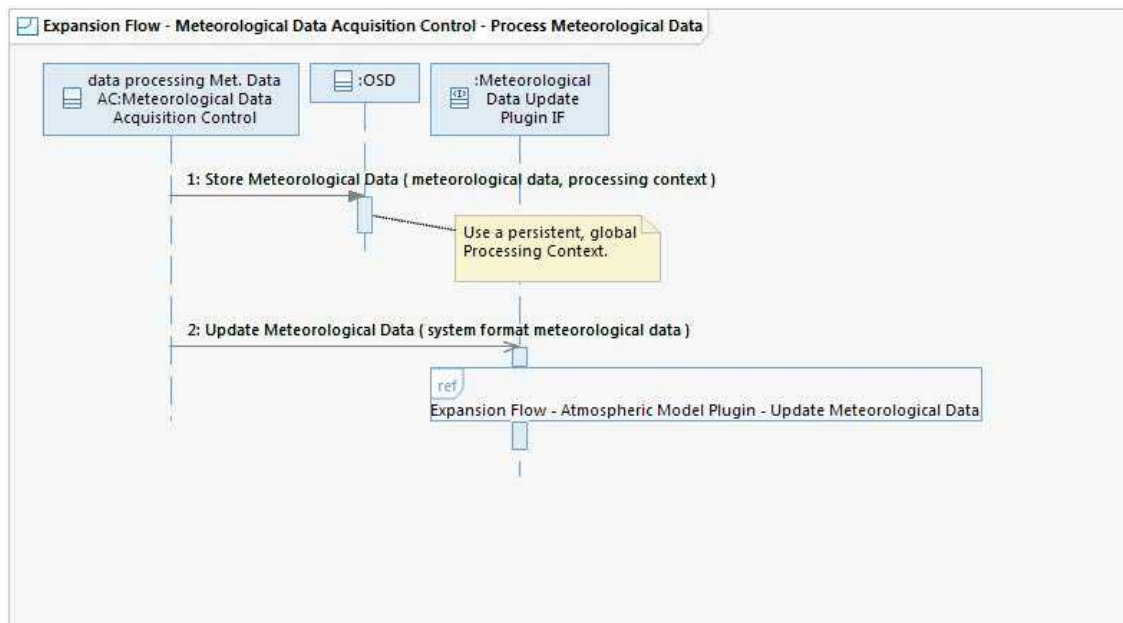
This flow shows how the Meteorological Data Acquisition Control responds to a callback from the System Clock to acquire, convert, and merge meteorological data before requesting a transfer of the data to the Data Processing Partition via the Transfer Control. Note that the Meteorological Data Acquisition Control in this flow runs only on the Data Acquisition Partition. The converted meteorological data is also stored on the data acquisition partition to facilitate forwarding to other Subsystems, External Data Centers, and Authorized External Users (see ‘System Automatically Distributes Data’ UCR).

## Operation Descriptions

None



## ***Expansion Flow - Meteorological Data Acquisition Control - Process Meteorological Data***



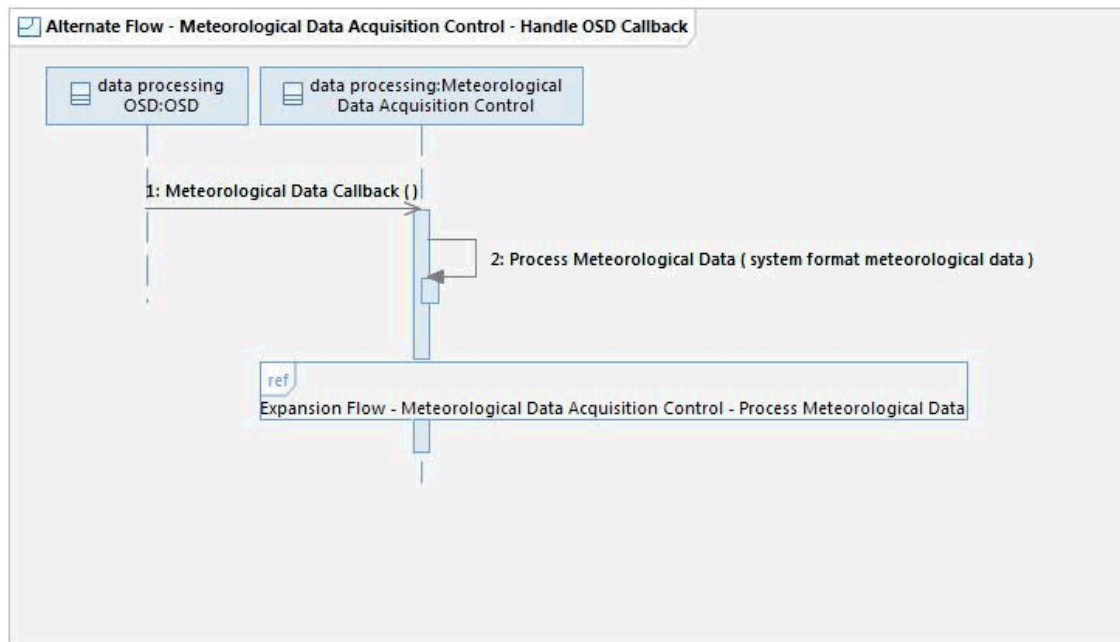
This flow shows Meteorological Data Acquisition Control on the Data Processing Partition processing System Format Meteorological Data. Meteorological Data Acquisition Control stores the new data in the OSD and updates an Atmospheric Model Plugin with the data via the Meteorological Data Update Plugin IF.

### **Operation Descriptions**

#### ***Operation: Meteorological Data Update Plugin IF::Update Meteorological Data()***

Provides new meteorological data to an Atmospheric Model Plugin. The plugin applies gravity corrections if required to the model at this time. The plugin is responsible for storing the updated atmospheric model in the OSD.

### *Alternate Flow - Meteorological Data Acquisition Control - Handle OSD Callback*

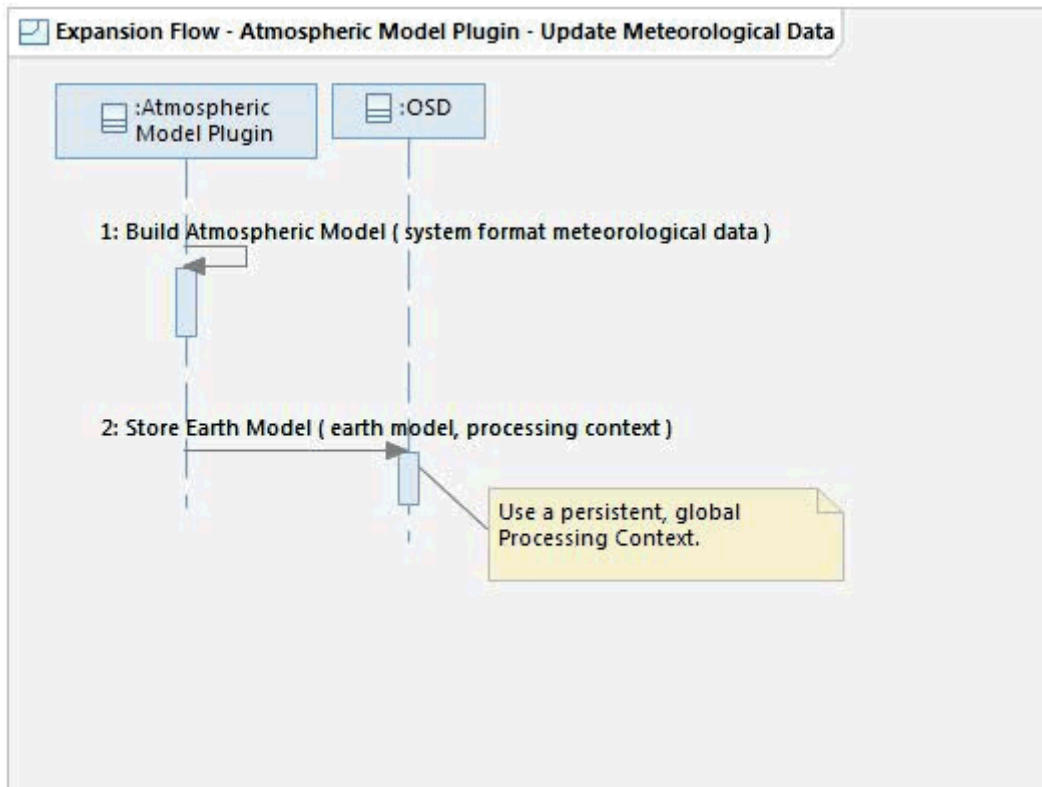


This flow depicts how the Meteorological Data Acquisition Control handles a callback from the OSD to process meteorological data that has been stored on the Data Processing Partition.

#### **Operation Descriptions**

None

### *Expansion Flow - Atmospheric Model Plugin - Update Meteorological Data*

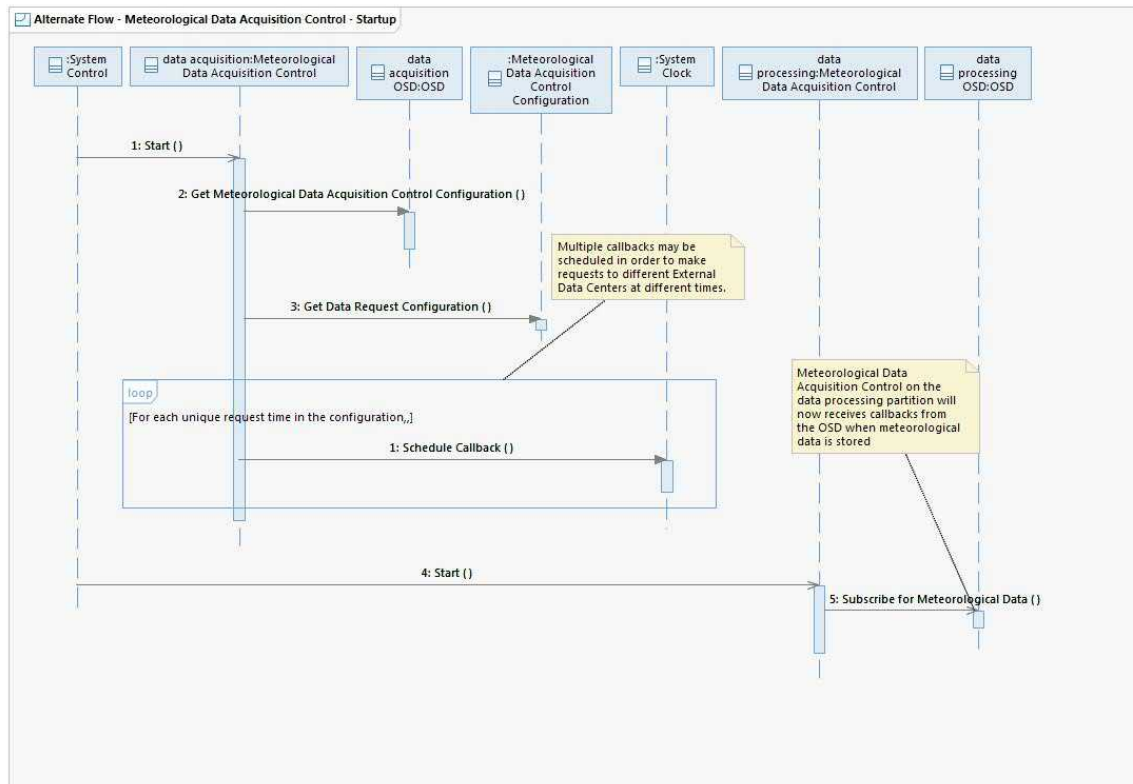


This is a notional flow showing an Atmospheric Model Plugin implementation is responsible for updating its atmospheric model (which it exposes as a realization of the Earth Model Plugin IF interface) with new meteorological data and storing the updated model in the OSD.

### **Operation Descriptions**

None

## Alternate Flow - Meteorological Data Acquisition Control - Startup



This flow shows how Meteorological Data Acquisition Control starts when requested by System Control. Meteorological Data Acquisition Control on the Data Acquisition Partition uses the OSD to get configuration describing from which External Data Centers it will request meteorological data and how often to make those requests. Meteorological Data Acquisition Control registers for System Clock callbacks at the time intervals it will request data. Meteorological Data Acquisition Control on the Data Processing Partition does not schedule callbacks for requesting meteorological data, but instead subscribes for OSD callbacks when meteorological data is stored to the OSD on the Data Processing Partition.

### Operation Descriptions

None

### State Machine Diagrams

None

### SSD Mappings

#### General:

**S-5623:** [*Threshold*] The system shall request global meteorological data (3D wind, temperature and uncertainties) from specified sources spanning the altitude range of 0 km (sea level) to 140

km above sea level.

**S-5624:** [*Threshold*] The system shall convert global meteorological data received in pressure coordinate format into altitude coordinate format.

**S-5652:** [*Extensibility*] The System shall compute corrections to wind velocity predictions based on a model for atmospheric gravity waves.

**S-5653:** [*Extensibility*] The System shall compute corrections to atmospheric temperature predictions based on a model for atmospheric gravity waves.

**S-5654:** [*Threshold*] The system shall compute an infrasound propagation model using gravity wave corrected wind velocity and atmospheric temperature predictions.

**S-5656:** [*Objective / Priority 1*] The system shall compute an infrasound propagation model that incorporates high resolution meteorological data.

**S-5714:** [*Threshold*] The System shall store global meteorological data.

**S-5717:** [*Extensibility*] The System shall store gravity wave corrections to temperature predictions.

**S-5759:** [*Objective / Priority 1*] The System shall request global meteorological data from specified sources within 10 minutes of its availability.

## Notes

### General:

1. One example of a meteorological format that the External Meteorological Data class could represent is GRIB2. GRIB2 is a file format used for storage and transport of gridded meteorological data. For more information about GRIB2 see:  
[https://weather.gc.ca/grib/what\\_is\\_GRIB\\_e.html](https://weather.gc.ca/grib/what_is_GRIB_e.html).
2. Different data items are stored on different partitions of the System (data or acquisition) in this UCR. System Format Meteorological Data is stored on both the Data Acquisition Partition and Data Processing Partition. The atmospheric model is stored on the data processing partition only. The System does not store External Format Meteorological Data received by the System from External Data Centers.
3. UCR 01.06 "System Synchronizes Processing Results" handles the synchronization of the Atmospheric Model and System Format Meteorological Data between the Primary and Backup systems.
4. Depending on Transfer Control Configuration, Transfer Control will either automatically

retransmit missing meteorological data from the Data Acquisition Partition to the Data Processing Partition or notify the System Controller of the missing data so the System Controller can manually request a retransfer using "Alternate Flow - System Controller Requests Transfer of Missing Meteorological Data" in System Receives Station Data UCR.

5. This Use Case Realization uses some common behavior and flows from UCR 01.01, 'System Receives Station Data', for handling the transfer of data from the Data Acquisition Partition to the Data Processing Partition.

**IDC Specific:**

1. The IDC requests meteorological data from External Data Centers with frequency on the order of hours. The IDC converts the data to the System Format Meteorological Data and then provides it to subscribers (see 'System Automatically Distributes Data' UCR).

2. The description for the Main Flow says that "The converted meteorological data is also stored on the data acquisition partition to facilitate forwarding to other Subsystems, External Data Centers, and Authorized External Users. . . ". Are Authorized External Users included, or is it just other Subsystems and EDC's? Is meteorological data distributed to Authorized External Users?

# IDC Use Case Realization Report

## UCR-02.06 System Builds Event Using Signal Detections

### Use Case Description

This architecturally significant use case describes how the System uses signal detections and features measured from those signal detections to build single station event hypotheses, build network event hypotheses, and associate previously unassociated signal detections to existing event hypotheses.

To build an event hypothesis, the System associates signal detections from one or more seismic, hydroacoustic, or infrasound stations. The System builds event hypotheses meeting predefined event formation criteria and associates previously unassociated signal detections to existing event hypotheses using signal detections, feature measurements based on those signal detections (see 'System Measures Signal Features' UC), and single station signal detection groups. The System references empirical knowledge from past events and geophysical models when forming event hypotheses and when associating previously unassociated signal detections to existing event hypotheses. The System makes signal detection phase assignments using information available when considering signal detections in the context of event hypotheses. The System validates event hypotheses and phase assignments using empirical and geophysical model based parameters. The System computes quality metrics for all event hypotheses.

The System Maintainer configures the phase assignment parameters, signal detection association parameters, and event hypothesis quality metric parameters (see 'Configures Processing Components' UC). The Analyst has the option to select values for these parameters during interactive review (see 'Builds Event' UC).

This use case may initiate reprocessing of earlier analysis steps such as signal enhancement (see 'System Enhances Signals' UC), signal detection (see 'System Detects Signals' UC) and signal feature measurement (see 'System Measures Signal Features' UC) if an iterative, feedback-based processing sequence is implemented.

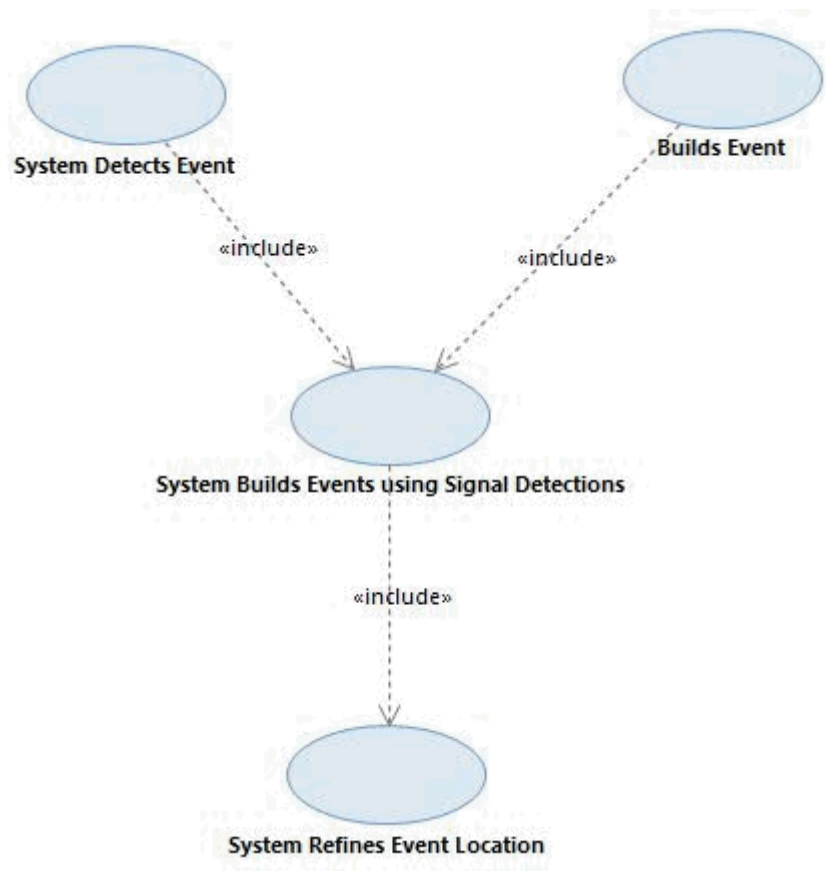
This use case is architecturally significant because it involves complex algorithms for automatically building and modifying events.

### Architecture Description

The Signal Detection Association Control class is responsible for controlling computations that associate signal detections to build new event hypotheses or update existing event hypotheses. Signal Detection Association Control may be invoked by Processing Sequence Control as part of executing a step in a Processing Sequence (see 'System Detects Event' UCR) or be manually

invoked by an Analyst as part of building an event (see 'Builds Events' UCR). Signal Detection Association Control uses a Signal Detection Associator Plugin to perform the association calculations. Multiple Signal Detection Associator Plugins exist in the system and each of them realizes a common plugin interface. The specific Signal Detection Associator Plugin varies dynamically at runtime based on the Signal Detection Association Parameters. When invoked from Processing Sequence Control, Signal Detection Association Control uses the Signal Detection Association Configuration to build the Signal Detection Association Parameters to pass to the Signal Detection Associator Plugin, selects and invokes the appropriate Signal Detection Associator Plugin based on those parameters, and stores the new or modified Events and Event Hypotheses via the OSD mechanism.

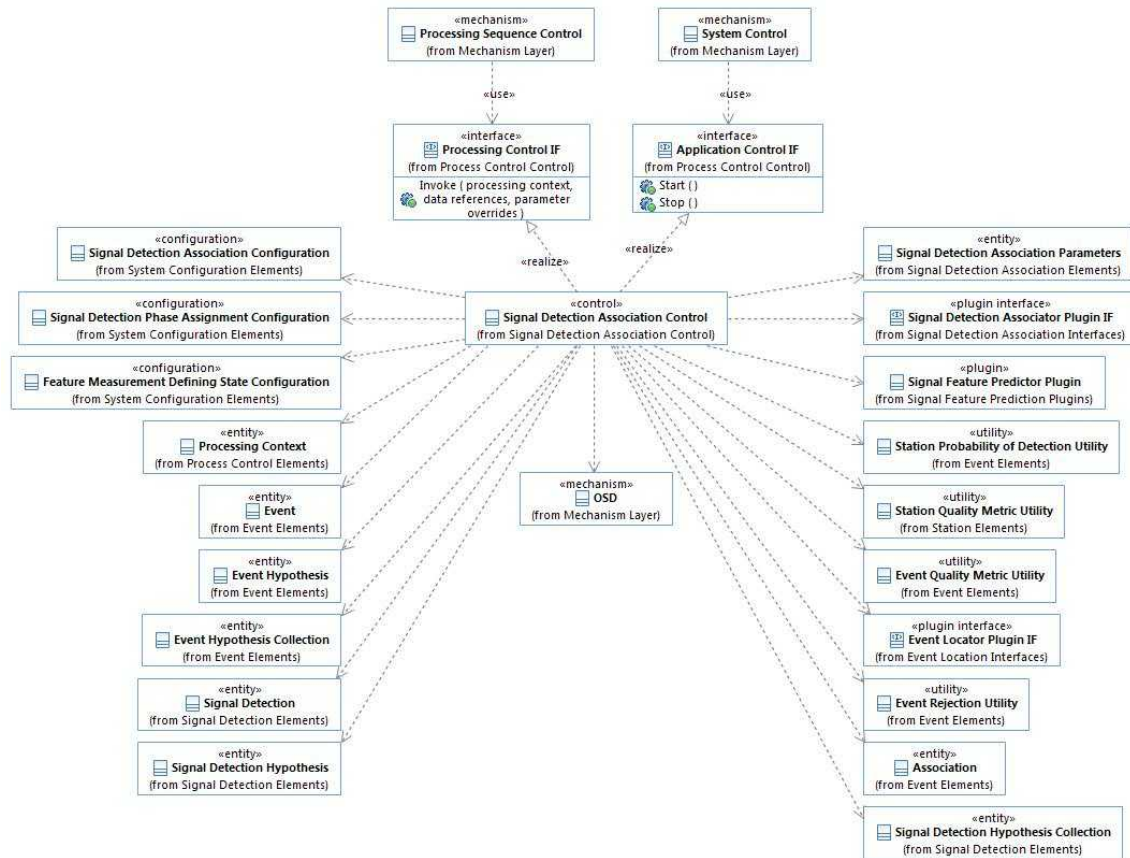
## Use Case Diagram





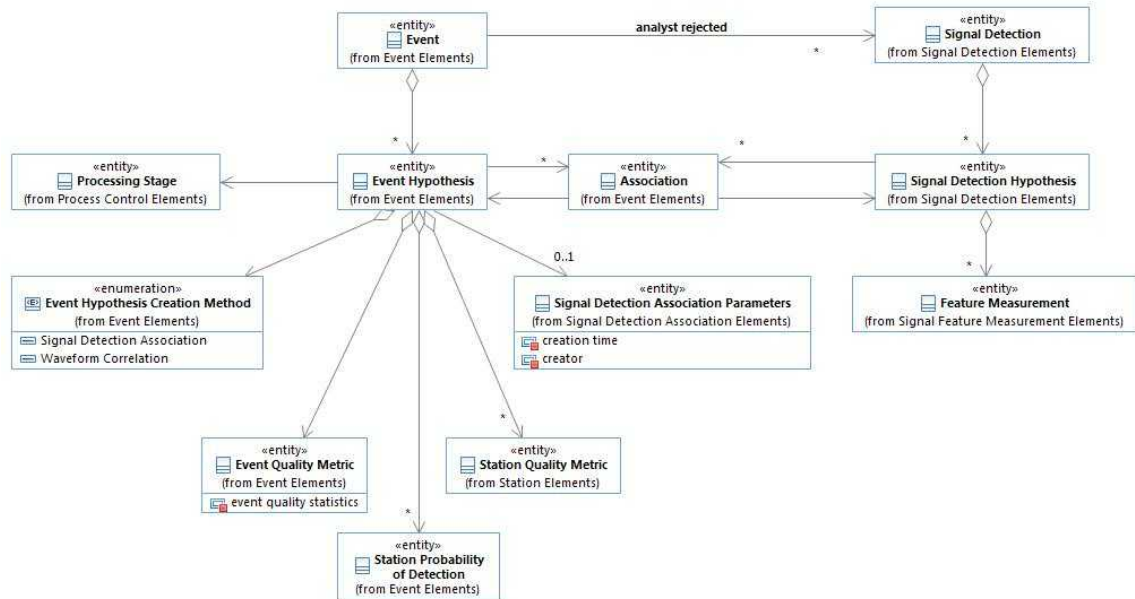
## Class Diagrams

### Classes - Signal Detection Association Control



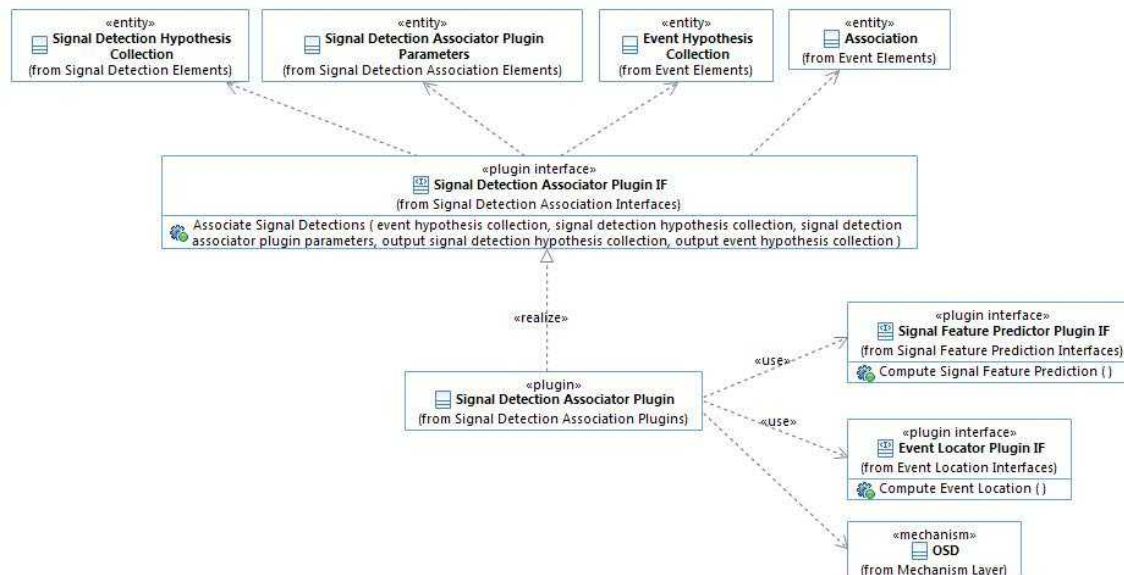
This diagram shows Signal Detection Association Control and related classes.

## Classes - Event Hypothesis



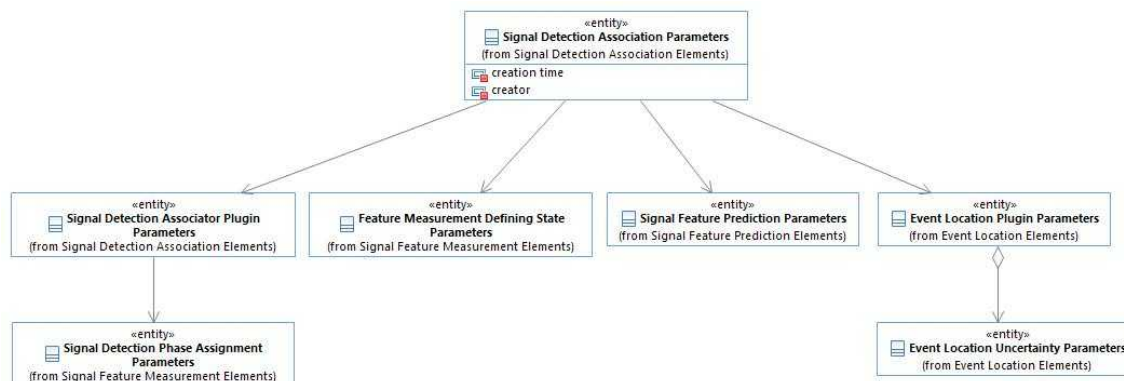
This diagram shows the Event Hypothesis class and related classes as used in this realization. Each Event Hypothesis has a creation method, quality metrics, and an association to the Signal Detection Association Parameters used by the Signal Detection Association Plugin that built the Event Hypothesis. These parameters track the Event Hypothesis' provenance. Signal Detection Hypothesis aggregates Feature Measurements, which Signal Detection Associator Plugins may use during association computations. Signal Detection Association Control uses the association between Event and Signal Detection to keep from reassociating a Signal Detection to an Event when an Analyst has previously unassociated the two.

### ***Classes - Signal Detection Associator Plugin***



This diagram shows the Signal Detection Associator Plugin, related elements, and related plugins. The Signal Detection Associator Plugin class shown on this diagram is a general class representing any possible implementation of the Signal Detection Associator Plugin IF. Depending on the algorithm implementation a Signal Detection Associator Plugin may access the OSD, use Signal Feature Predictor Plugins, and use Event Locator Plugins.

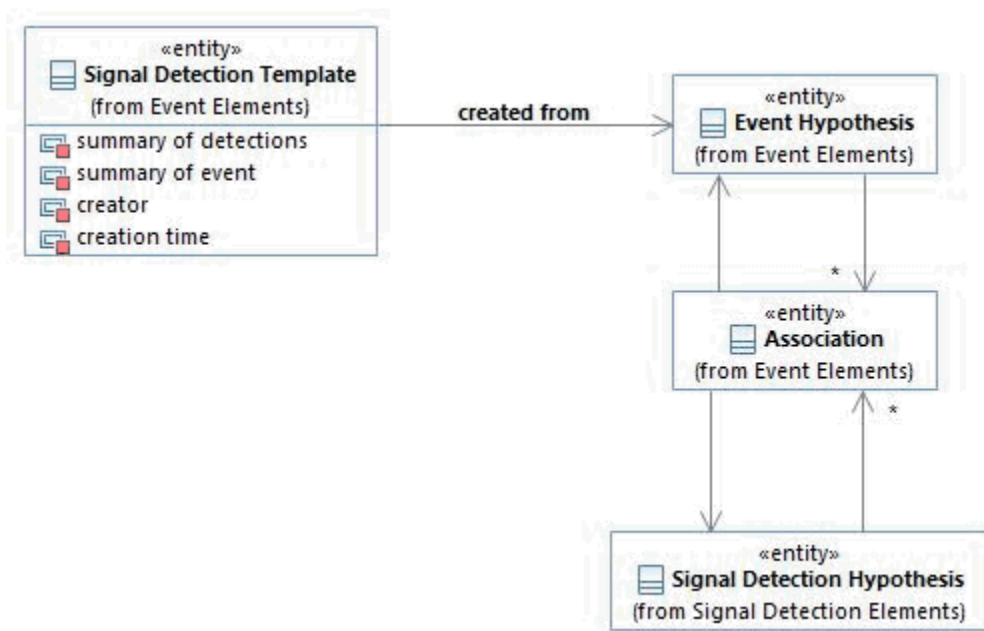
### *Classes - Signal Detection Association Parameters*



This diagram shows details of the Signal Detection Association Parameters class. Signal Detection Association Parameters includes the information Signal Detection Association Control uses to invoke a single Signal Detection Associator Plugin, to set feature measurement defining states, and to refine the locations of event hypotheses it creates and modifies. Signal Detection Associator Plugin Parameters collects the parameters used by plugins, including the plugin specific parameters and phase assignment parameters. Signal Detection Association Control also uses the phase assignment parameters to assign signal detection phase labels. Signal Detection Association Control creates these parameter classes based on the Signal Detection Association Configuration, Signal Detection Phase Assignment Configuration, and Feature Measurement

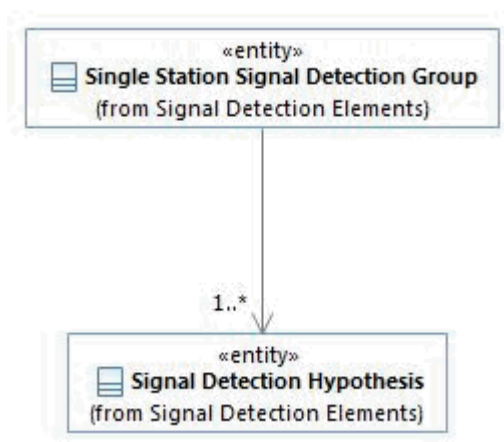
Defining State Configuration classes.

### ***Classes - Signal Detection Template***



This diagram shows details for the Signal Detection Template class. Signal Detection Associator Plugin implementations may use Signal Detection Templates to build new event hypotheses and associate additional detections to existing event hypotheses by using lists of signal detections matching the pattern of signal detections associated to the Signal Detection Template's event hypothesis.

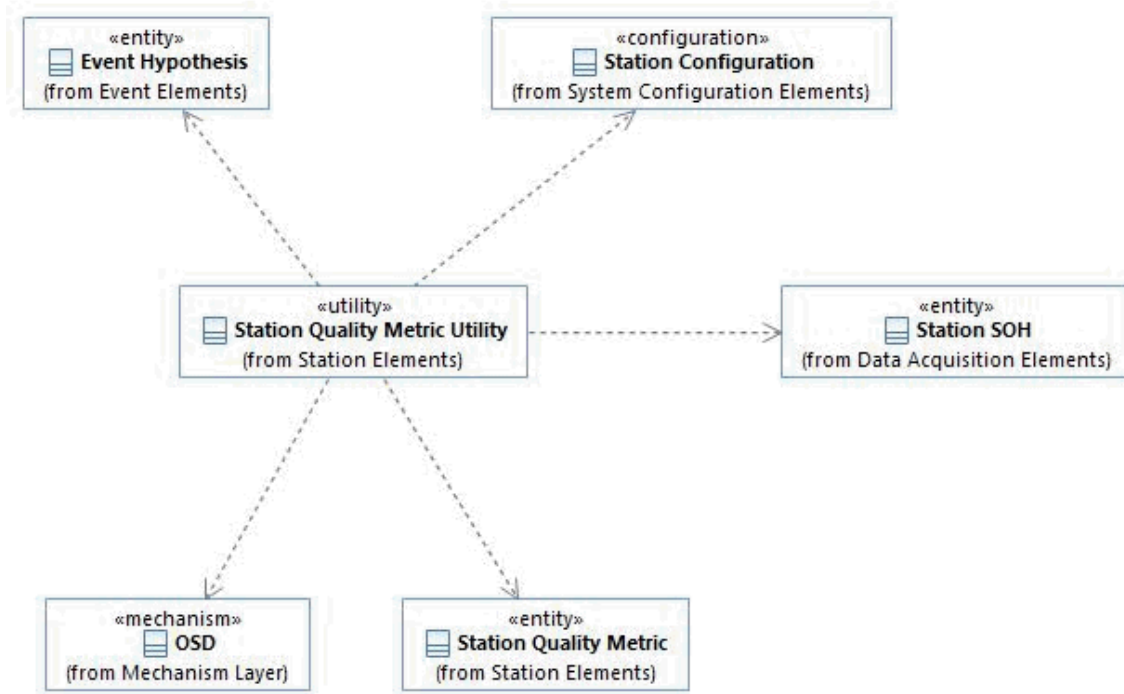
### ***Classes - Single Station Signal Detection Group***



This diagram shows the Single Station Signal Detection Group class. A Single Station Signal Detection Group groups multiple Signal Detection Hypotheses detected by the same station for the same Event. Station processing creates Single Station Signal Detection Groups (see 'System Measures Signal Features' UCR) and Signal Detection Associator Plugins may use the groups

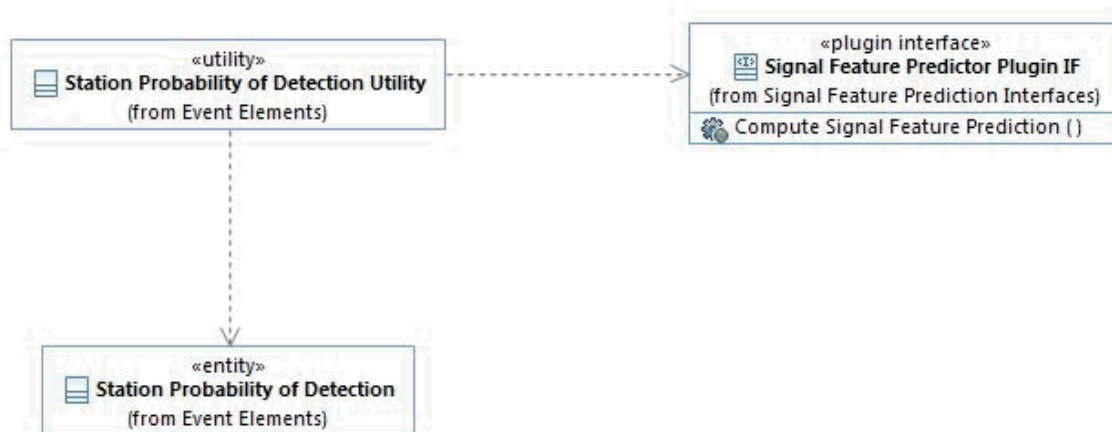
when creating Event Hypotheses.

### ***Classes - Station Quality Metric Utility***



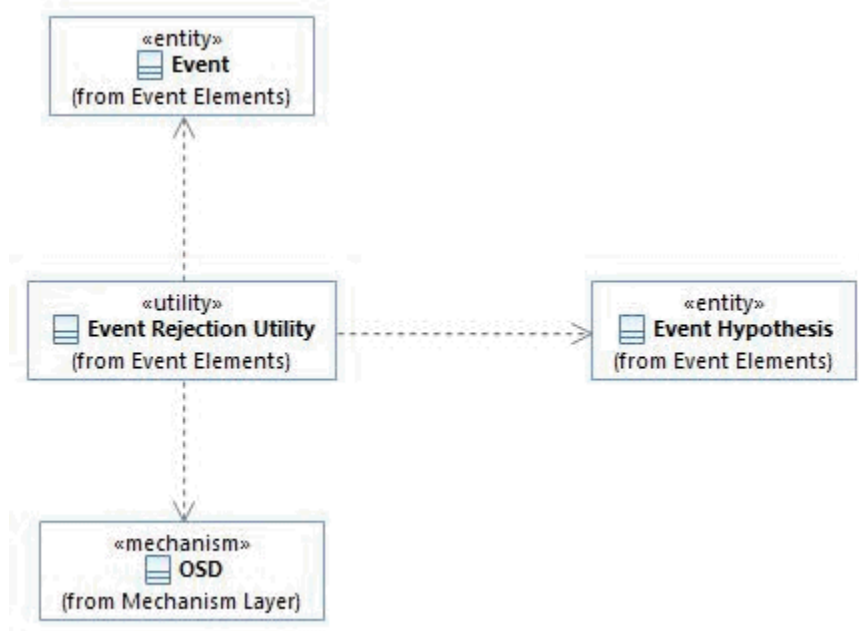
This diagram shows the Station Quality Metric Utility class and related classes it uses to compute Station Quality Metrics.

### ***Classes - Station Probability of Detection Utility***



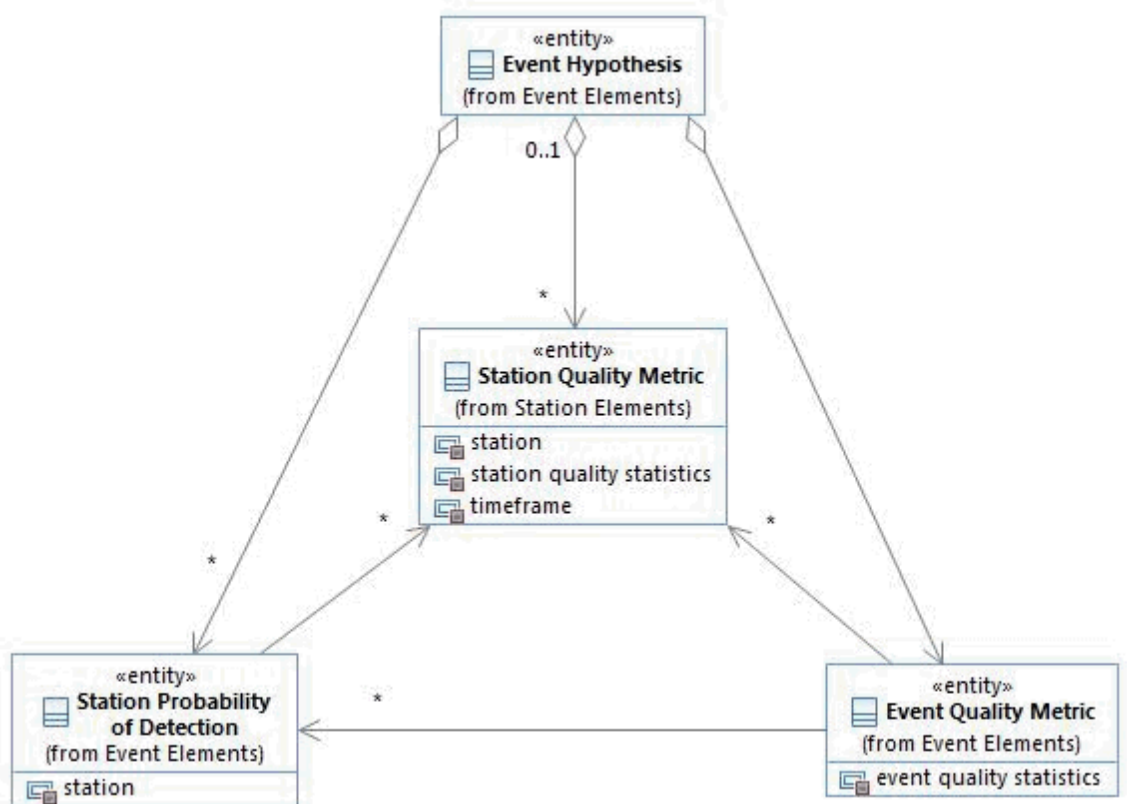
The diagram shows the Station Probability of Detection Utility and related classes. The utility uses a class realizing the Signal Feature Predictor Plugin IF interface to calculate a station's probability of detecting particular phases of an event. The utility uses these probabilities to determine the overall station probability of detecting an event, which is represented by the Station Probability of Detection class.

### *Classes - Event Rejection Utility*



This diagram shows the Event Rejection Utility class and related classes it uses to reject invalid events.

## Classes - Quality Metrics



This diagram shows the relationships between the Station Quality Metric, Station Probability of Detection, and Event Quality Metric classes. The Station Probability of Detection and Event Quality classes have associations to the Station Quality Metric class since it is an input to the calculations which create those classes. Similarly, Event Quality Metric is associated to Station Probability of Detection since that class is an input to event quality calculations.

## Class Descriptions

### **<<configuration>> Feature Measurement Defining State Configuration**

Represents all signal detection feature measurement defining state configuration in the system. This includes all configurations used to determine which signal detection feature measurements are by default defining and non-defining for various types of system calculations.

### **<<configuration>> Signal Detection Association Configuration**

Represents all signal detection association configuration in the system. Includes event location configuration used to refine the locations of event hypotheses created or modified during signal detection association. May also include associations to plugin-specific configurations for individual Signal Detection Associator Plugins (each Signal Detection Associator Plugin may define a plugin specific configuration class containing e.g. configuration for other plugins that plugin uses when it is invoked, single station event formation criteria configuration, network event formation criteria configuration, etc.)



**<<configuration>> *Signal Detection Phase Assignment Configuration***

Represents all signal detection phase assignment configuration in the system. This includes all station, channel, frequency, and geographic region specific configurations used to determine signal detection phase assignments.

**<<configuration>> *Station Configuration***

Represents the configurable aspects of a station such as its name, location, whether it is enabled, etc.

**<<control>> *Signal Detection Association Control***

Control class responsible for controlling signal detection association calculations. Retrieves configuration from the OSD, invokes the appropriate Signal Detection Associator Plugin, computes quality metrics, and stores the new or modified events in the OSD.

**<<entity>> *Association***

Represents an association between a Signal Detection Hypothesis and an Event Hypothesis.

**<<entity>> *Event***

Represents information about an Event. Keeps track of all the Event Hypotheses for the Event, which Event Hypothesis is the preferred one for each processing stage, the active analysts for the Event (i.e. whether the Event is under "active review"), whether the Event is "complete" for each processing stage, and other Event-related information.

**<<entity>> *Event Hypothesis***

Represents geophysical information about an Event as determined by an Analyst or through pipeline processing. There can be multiple Event Hypotheses for the same Event (e.g. different associated Signal Detection Hypotheses, different location solutions).

**<<entity>> *Event Location Plugin Parameters***

Represents all parameters passed to an Event Locator Plugin. This includes parameters describing default feature measurement defining states and the types of uncertainty bounds the plugin should compute. May also include parameters specific to the plugin being invoked.

**<<entity>> *Event Location Uncertainty Parameters***

Represents the type of uncertainty bound (Confidence, Coverage or K-Weighted), confidence level and scaling factor for the locator to use when computing event location uncertainty.

**<<entity>> *Event Quality Metric***

Represents the quality of a single event hypothesis. Event quality is based on a variety of event quality statistics, possibly including the Station Quality Metric and Station Probability of Detection for Stations with Signal Detections associated to the Event Hypothesis as well as for Stations that did not detect the Event Hypothesis, the algorithms used to detect the event, event location and event location uncertainty, etc.



### **<<entity>> *Feature Measurement***

Represents the value and uncertainty of a measured feature of a signal detection.

### **<<entity>> *Feature Measurement Defining State Parameters***

Represents defining state parameters for feature measurements. The parameters include the following for each feature measurement for each type of calculation (e.g. location, magnitude, etc.)

- Whether the feature measurement is initially defining or non-defining
- Whether an algorithm is free to toggle the defining state

The analyst can override the defining/non-defining state for these parameters, unless prohibited by the default defining/non-defining state:

- Time, azimuth, or slowness measurement of a signal detection for event hypothesis relocation
- Signal Detection measurements for event hypothesis relocation based on channel
- Signal Detection measurements for event hypothesis relocation based on signal detection phase assignment

### **<<entity>> *Processing Context***

Represents the context in which data is being stored and/or processed. This includes the Processing Stage (either automatic or interactive) and Interval performing the processing session (e.g. processed by Analyst vs. processed by System). For Analyst processing, may identify the Analyst work session. For System processing, may identify the Processing Sequence and/or Processing Step being executed (including a way to identify a particular Processing Sequence and Processing Step among the many possible instantiations), the visibility for the results (private vs. global), and the lifespan of the data (transient vs. persistent). This information is needed by the Processing Sequence Control to manage the execution of Processing Sequences, which may execute in the context of an Analyst refining an Event or in the context of the system initiating automatic processing. It is also needed by the Object Storage and Distribution (OSD) mechanism to determine how to store and distribute the data.

### **<<entity>> *Processing Stage***

Represents a named stage of data processing, which may be part of the System Maintainer-defined workflow or an Analyst-defined stage outside the workflow. All Processing Results are associated to a Processing Stage. The previous processing stage indicates the stage to be used as the default starting point when creating new processing results in the stage (e.g. when refining an event in the stage).

### **<<entity>> *Signal Detection***

Represents information about a Signal Detection and keeps track of all the Signal Detection Hypotheses for the Signal Detection. Represents information about a Signal Detection and keeps track of all the Signal Detection Hypotheses for the Signal Detection. For an unassociated Signal Detection the preferred Signal Detection Hypothesis is the most recently created Signal Detection Hypothesis. For an associated Signal Detection the preferred Signal Detection Hypothesis is the one associated to a preferred Event Hypothesis.

**<<entity>> *Signal Detection Association Parameters***

Represents the parameters used by a particular invocation of Signal Detection Association Control. This includes parameters used to invoke a Signal Detection Associator Plugin to build events and parameters used to invoke a Event Locator Plugin to refine event locations as well as parameters controlling additional calculations performed in the control class.

**<<entity>> *Signal Detection Associator Plugin Parameters***

Represents the parameters used by a call to a Signal Detection Associator Plugin. This includes parameters that apply to all Signal Detection Associator Plugins and may include plugin specific parameters.

**<<entity>> *Signal Detection Hypothesis***

Represents geophysical information about a Signal Detection as determined by an Analyst or through pipeline processing. There can be multiple Signal Detection Hypotheses for the same Signal Detection (e.g. different onset times, different phase labels).

**<<entity>> *Signal Detection Template***

A template that represents the pattern of Signal Detections for an Event (i.e. channels detected, relative positions for each detection, phases, etc.). An Analyst may apply the template to quickly build new Events that match the pattern of detections. Signal Detection Associator Plugins (see “System Builds Events using Signal Detections” UCR) may also use Signal Detection Templates to build new Event Hypotheses and associate additional detections to Event Hypotheses matching the template. Also includes summary information about the original Event from which the template was created (e.g. Event location, magnitude, etc.), as an aid to the Analyst in finding and applying a relevant template.

**<<entity>> *Single Station Signal Detection Group***

Represents groups of Signal Detection Hypotheses detected by the same station for a single Event. Station processing creates Single Station Signal Detection Groups to help assign initial phase labels for the detections (see “System Measures Signal Features” UCR) and network processing uses the groups to help create Event Hypotheses (see “System Builds Events using Signal Detections” UCR).

**<<entity>> *Station Probability of Detection***

Represents a Station’s probability of detecting an Event Hypothesis.

**<<entity>> *Station Quality Metric***

Represents a Station’s quality for a particular time. Separate station quality metrics can be computed for a Station with each metric based on different selections of the Station’s raw and derived waveforms (e.g. Station Quality Metric could be computed using Waveforms from a Station’s raw Channels and a separate quality metric could be computed for a beam created from the Station’s waveforms).

**<<entity>> *Station SOH***

Represents the state-of-health for a single station. Includes state-of-health information acquired

from the Station and digital signature authentication status.

**<<enumeration>> *Event Hypothesis Creation Method***

Indicates the type of algorithm used to create an event hypothesis.

**<<interface>> *Application Control IF***

Defines the interface implemented by all <<control>> classes in the system that are controlled by System Control.

**<<interface>> *Processing Control IF***

Defines the interface implemented by all <<control>> classes in the system that are controlled by the Processing Sequence Control <<mechanism>>. <<control>> classes realize this common interface to support configurable processing sequence definition and execution. Processing Sequence Control uses the Invoke() operation declared in Processing Control IF to call <<control>> classes while executing processing sequences. When called in this way the <<control>> classes operate on the provided data (e.g. event hypotheses, signal detections, etc.) using either default parameters configured by the System Maintainer and loaded by the <<control>> class on startup or override parameters provided to the Invoke() operation.

**<<mechanism>> *OSD***

Represents the Object Storage and Distribution mechanism for storing and distributing data objects internally within the system.

**<<mechanism>> *Processing Sequence Control***

Mechanism for executing and controlling processing sequences configured by the System Maintainer.

**<<plugin interface>> *Event Locator Plugin IF***

Standard interface for all Event Locator plugins. All Event Locator plugins in the system realize this interface.

**<<plugin interface>> *Signal Detection Associator Plugin IF***

Standard interface for all Signal Detection Associator plugins. All Signal Detection Associator plugins in the system realize this interface.

**<<plugin interface>> *Signal Feature Predictor Plugin IF***

Standard interface for all Signal Feature Predictor plugins. All Signal Feature Predictor plugins in the system realize this interface. Plugins that implement Signal Feature Predictor IF may predict different types of signal features, such as travel time, azimuth, slowness, amplitude, and probability of detection.

**<<plugin>> *Signal Detection Associator Plugin***

An abstract base class that realizes the Signal Detection Associator Plugin IF interface. Specializations of this class may define specific configurations and parameters used in their implementations, such as parameters for Signal Feature Predictor Plugins, single station event

formation criteria parameters, or network event formation criteria parameters.

### <<plugin>> *Signal Feature Predictor Plugin*

Abstract class that represents any/all of the Signal Feature Predictors that may be plugged in to the system behind the Signal Feature Predictor IF plugin interface. Signal Feature Predictors are responsible for calculating signal feature predictions.

### <<utility>> *Event Quality Metric Utility*

Utility class that computes the event quality metric. The quality metric may be based on how well an event hypothesis meets the event definition criteria, the Event Hypothesis Creation Method, etc.

### <<utility>> *Event Rejection Utility*

Utility class used to reject event hypotheses.

### <<utility>> *Station Probability of Detection Utility*

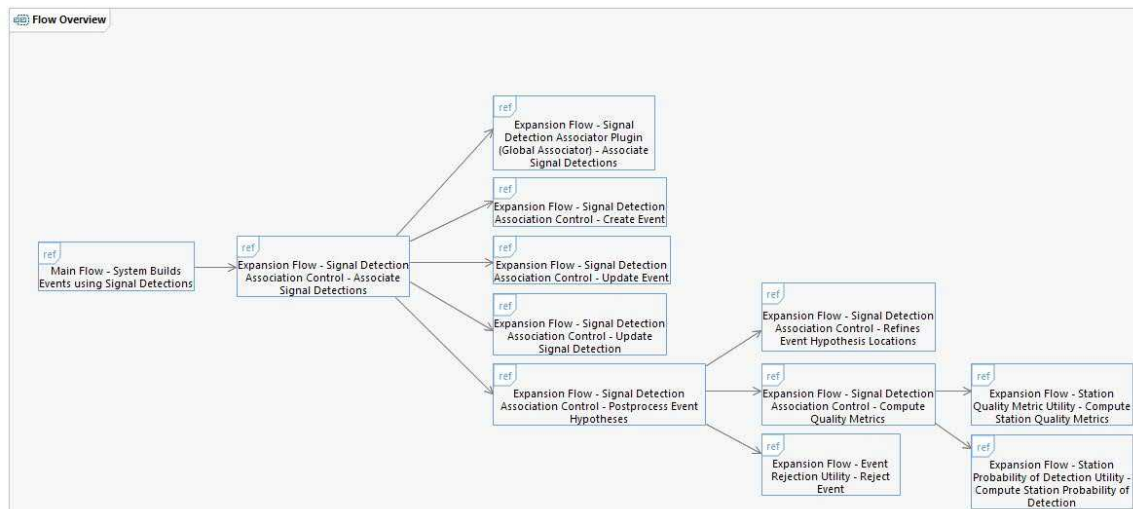
Utility that computes station probability of detecting events.

### <<utility>> *Station Quality Metric Utility*

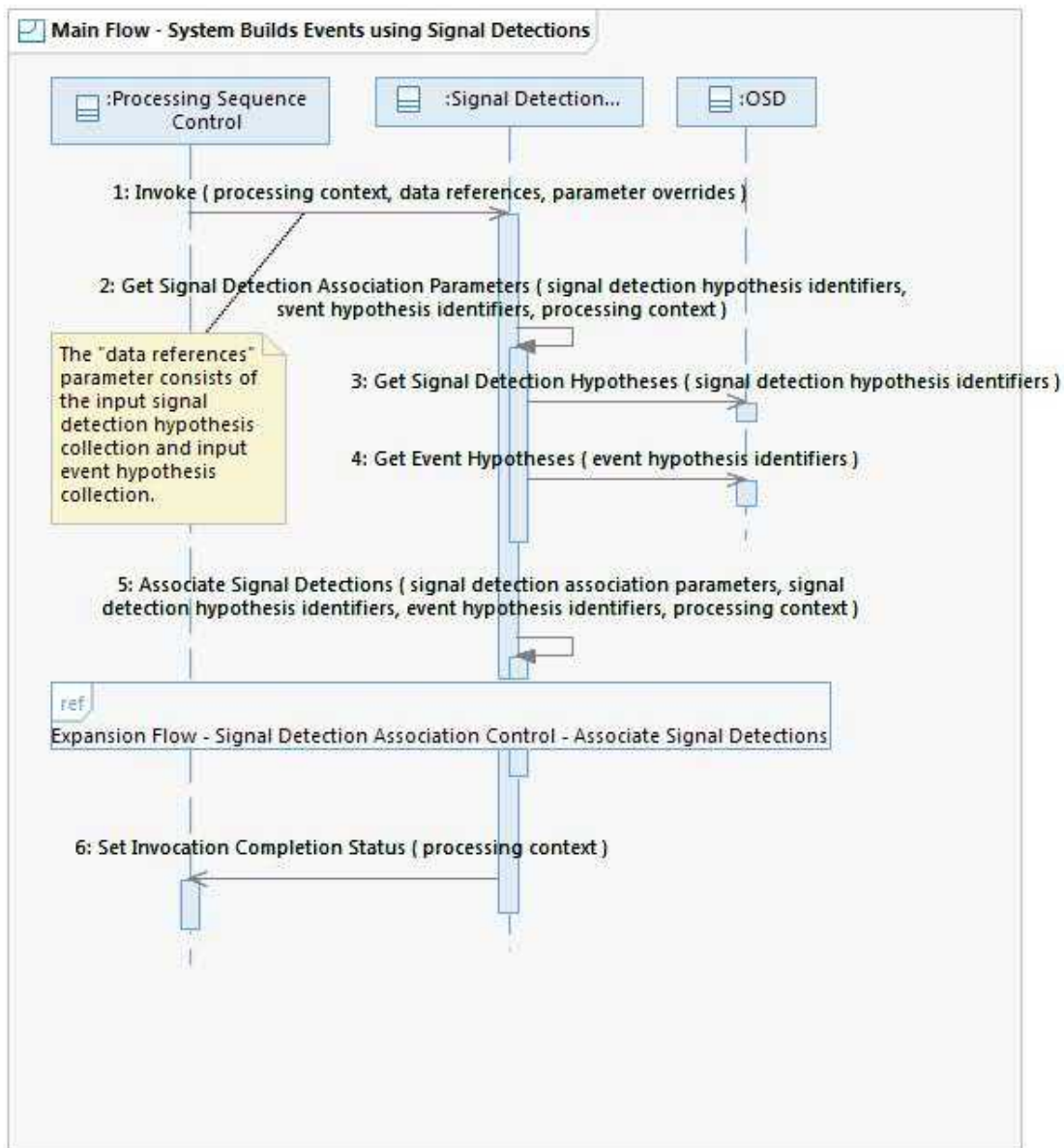
Utility class that computes the station quality metric. Computes the metric for stations using waveforms, SOH information, etc. and also computes the metric for configured derived channels (e.g. detection beams). Can compute a continuous station quality metric for use in performance monitoring and a different but related station quality metric based on an event hypothesis.

## Sequence Diagrams

### Flow Overview



## Main Flow - System Builds Events using Signal Detections



This flow shows how the system builds events by associating signal detections. This flow is stimulated by the Processing Sequence Control mechanism as part of executing an automatic processing sequence. Processing Sequence Control populates the data references parameter with a signal detection hypothesis collection and an event hypothesis collection. The signal detection hypothesis collection contains the signal detection hypotheses that this UCR uses to build new event hypotheses and to update existing event hypotheses. Depending on the Processing Configuration that Processing Sequence Control used to invoke this UCR, this collection may contain any combination of unassociated signal detections that have never been reviewed by an Analyst, unassociated signal detections that have previously been reviewed by an Analyst, signal detections associated to events that have not been reviewed by an Analyst, or signal detections

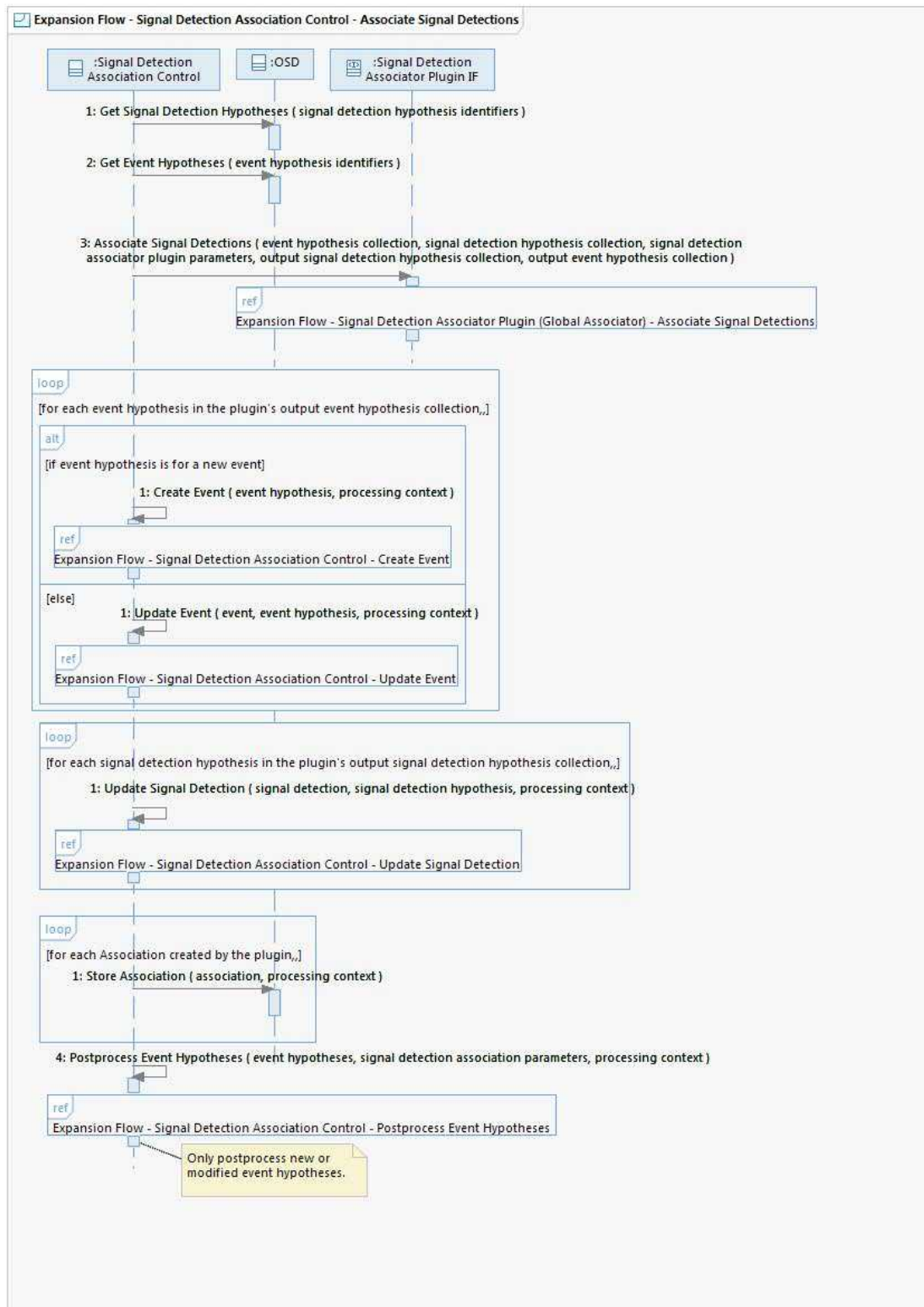
associated to Analyst reviewed events. The event hypothesis collection contains existing event hypotheses whose signal detection associations may be updated by this UCR. It is possible for this collection to contain event hypotheses previously reviewed by an Analyst.

The System Maintainer configures the conditions in which Processing Sequence Control executes these sequences (see 'Defines Processing Sequence' UCR). For more information about the Processing Sequence Control mechanism see 'System Detects Event' UCR.

### **Operation Descriptions**

None

## Expansion Flow - Signal Detection Association Control - Associate Signal Detections



This flow shows how Signal Detection Association Control associates signal detections to build new event hypotheses and to update existing event hypotheses. Signal Detection Association Control invokes a Signal Detection Associator Plugin to perform the association calculations. The control class uses Event Hypotheses output by the plugin to create new Events or to update existing Events and then postprocesses the event hypotheses to prepare for further processing by other UCRs. Signal Detection Association Control also stores any modified Signal Detection Hypotheses in the OSD.

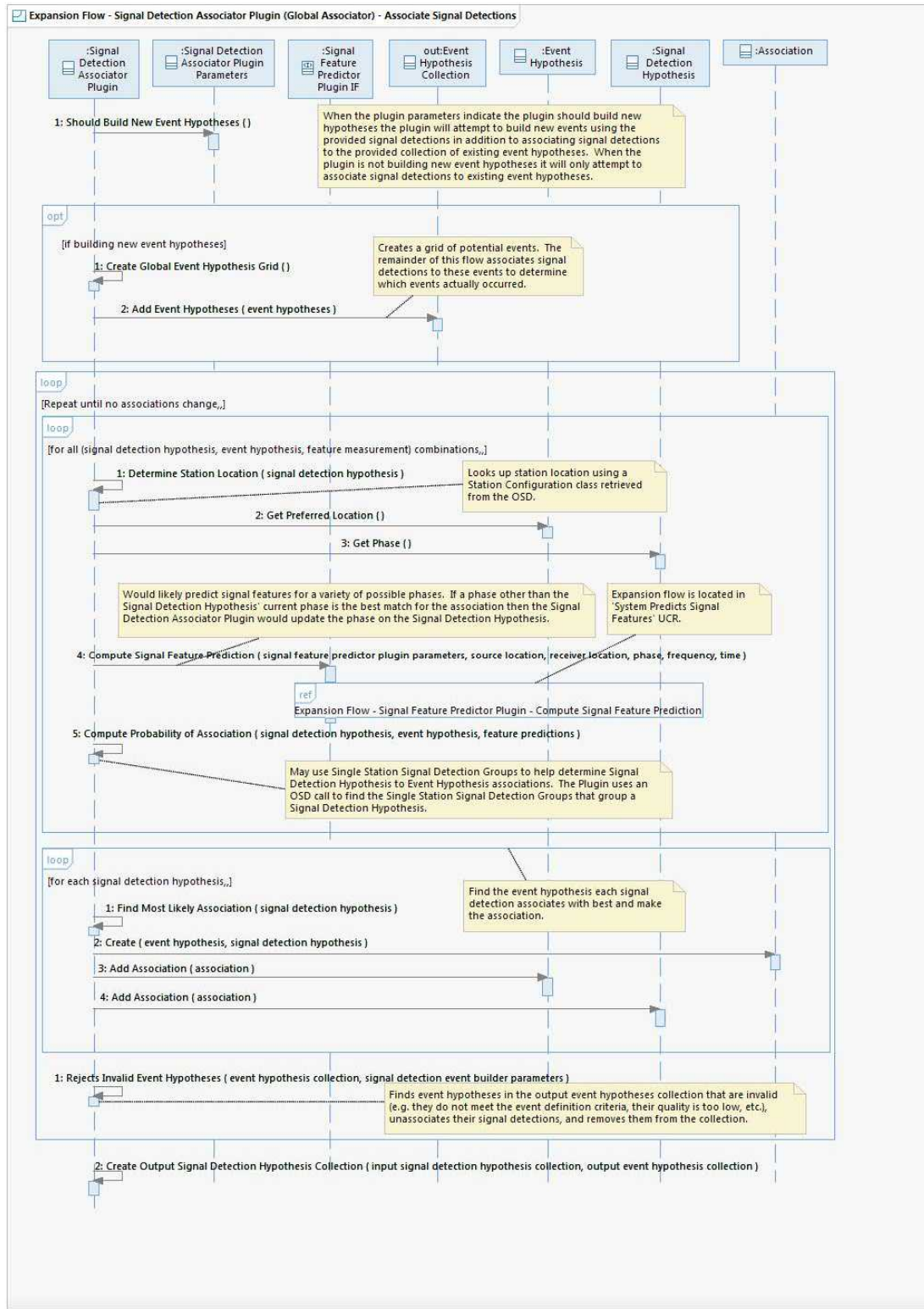
## **Operation Descriptions**

### ***Operation: Signal Detection Associator Plugin IF::Associate Signal Detections()***

Classes realizing the Signal Detection Associator Plugin IF interface implement this operation to build new event hypotheses, associate signal detection hypotheses to existing event hypotheses, or both. The input signal detection hypothesis collection contains all of the signal detection hypotheses the plugin may associate to events and the input event hypothesis collection contains all of the existing event hypotheses that may be associated to the input signal detections. The output signal detection hypothesis collection contains the signal detection hypotheses from the input collection that were updated by this operation (e.g. new phase assignment, associated to an event hypothesis) as a result of this operation. The output event hypothesis collection contains all of the event hypotheses that were modified or created by this operation.



## Expansion Flow - Signal Detection Associator Plugin (Global Associator) - Associate Signal Detections

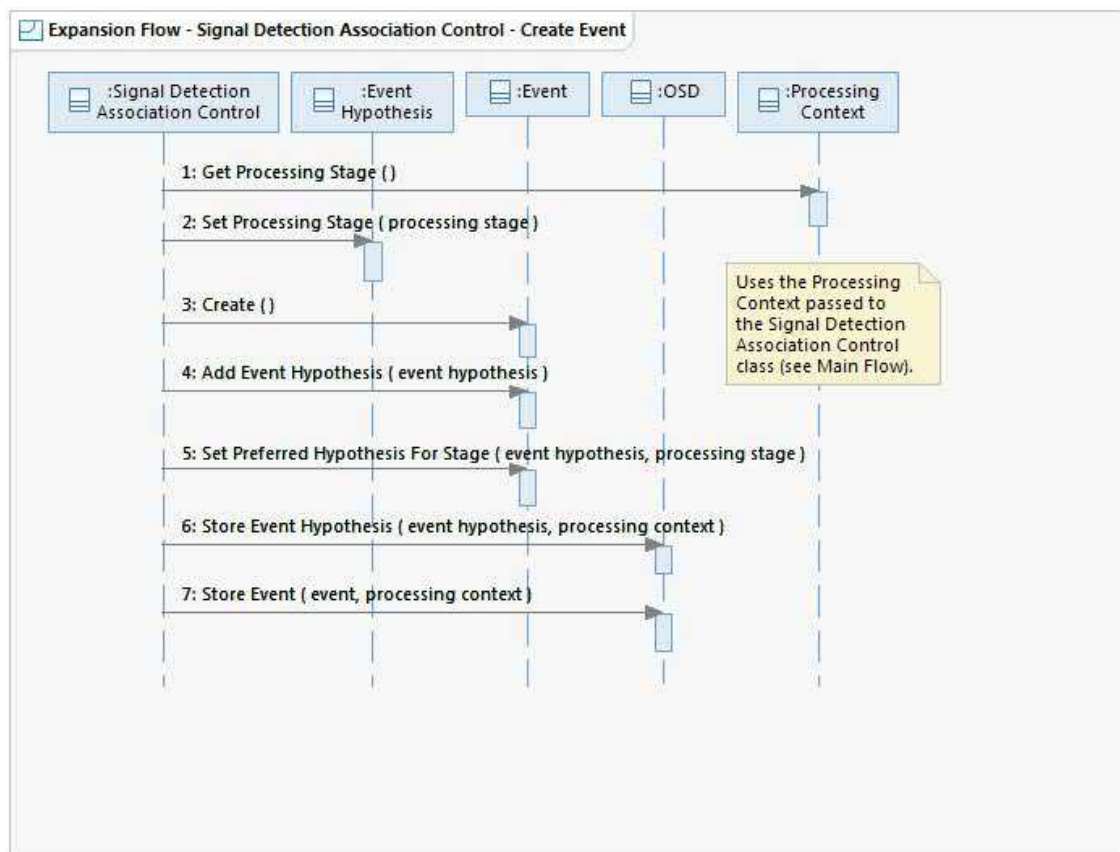


This flow is a notional flow showing how a Signal Detection Associator Plugin may associate signal detections to build new event hypotheses and to update the associations to existing event hypotheses. Other plugins may exist which use different algorithms (e.g. associate signal detections by matching the signal detection pattern from a waveform correlation reference event, associate signal detection by matching a Signal Detection Template, etc.). The plugin creates an Event Hypothesis grid of potential events and associates Signal Detections to Events based on the probability of signal detection to event association. This is an iterative process that continues until no associations change. At the end of each iteration the plugin removes improbable events from the event hypothesis grid, freeing the associated signal detections for association to other events.

## Operation Descriptions

None

### *Expansion Flow - Signal Detection Association Control - Create Event*



This flow shows Signal Detection Association Control creating a new Event, adding an Event Hypothesis as the preferred hypothesis of that event, and saving both in the OSD.

## Operation Descriptions

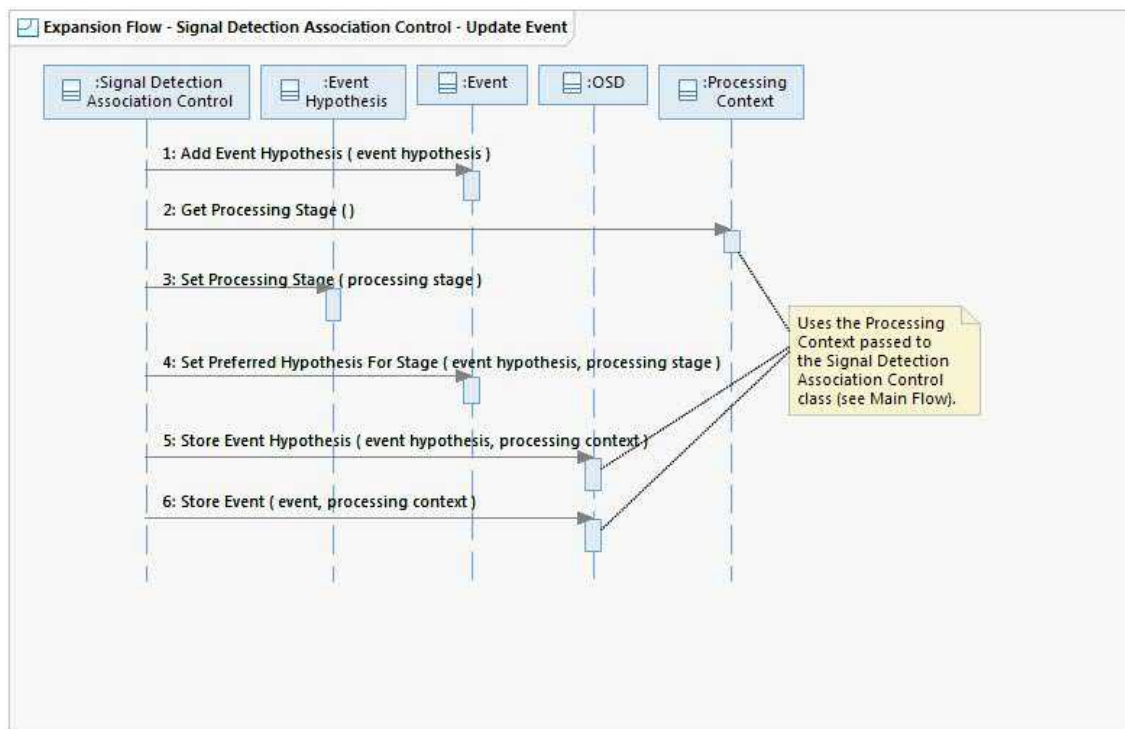
### *Operation: OSD::Store Event Hypothesis()*

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

**Operation: *OSD::Store Event()***

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

**Expansion Flow - Signal Detection Association Control - Update Event**



This flow shows Signal Detection Association Control adding a new Event Hypothesis to an Event, setting the hypothesis as the preferred hypothesis for that event, and saving both in the OSD.

**Operation Descriptions**

**Operation: *OSD::Store Event Hypothesis()***

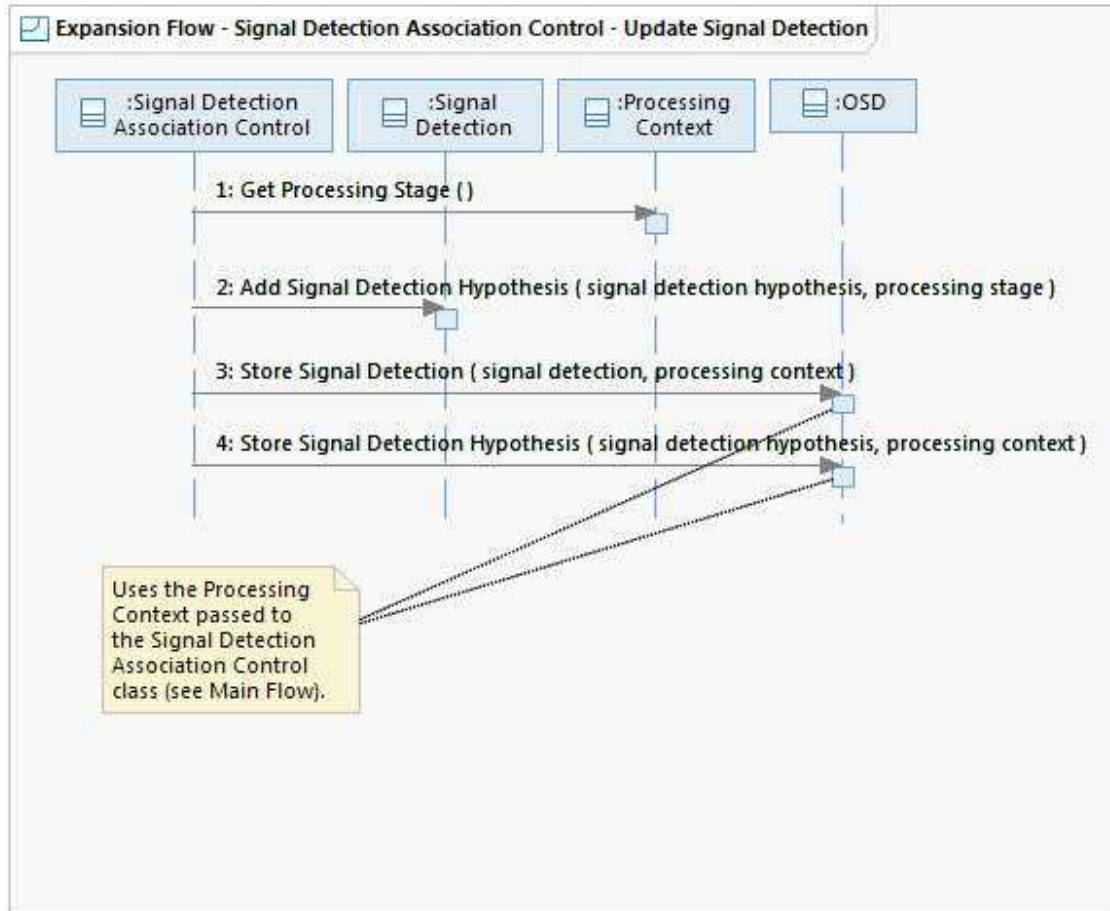
Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

**Operation: *OSD::Store Event()***

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via

callbacks.

### ***Expansion Flow - Signal Detection Association Control - Update Signal Detection***



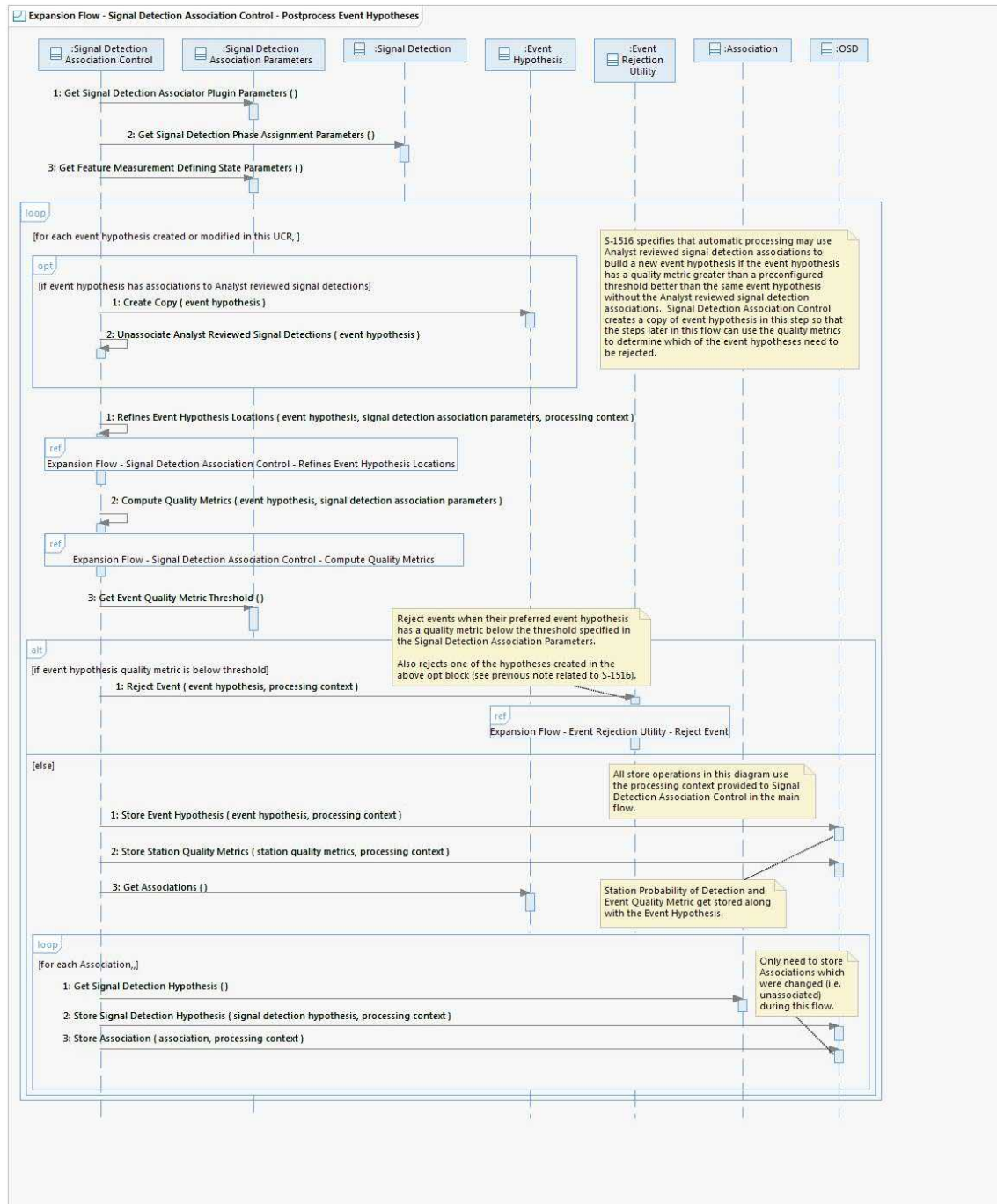
This flow shows Signal Detection Association Control adding a new Signal Detection Hypothesis to a Signal Detection and storing both of them in the OSD.

### **Operation Descriptions**

#### ***Operation: OSD::Store Signal Detection()***

Store the given Signal Detection with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

## Expansion Flow - Signal Detection Association Control - Postprocess Event Hypotheses



This flow shows Signal Detection Association Control refining the location of each event hypothesis created or modified in this UCR, computing quality metrics for those event hypotheses, and rejecting any hypotheses not meeting a preconfigured minimum quality metric.

## Operation Descriptions

### ***Operation: OSD::Store Event Hypothesis()***

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

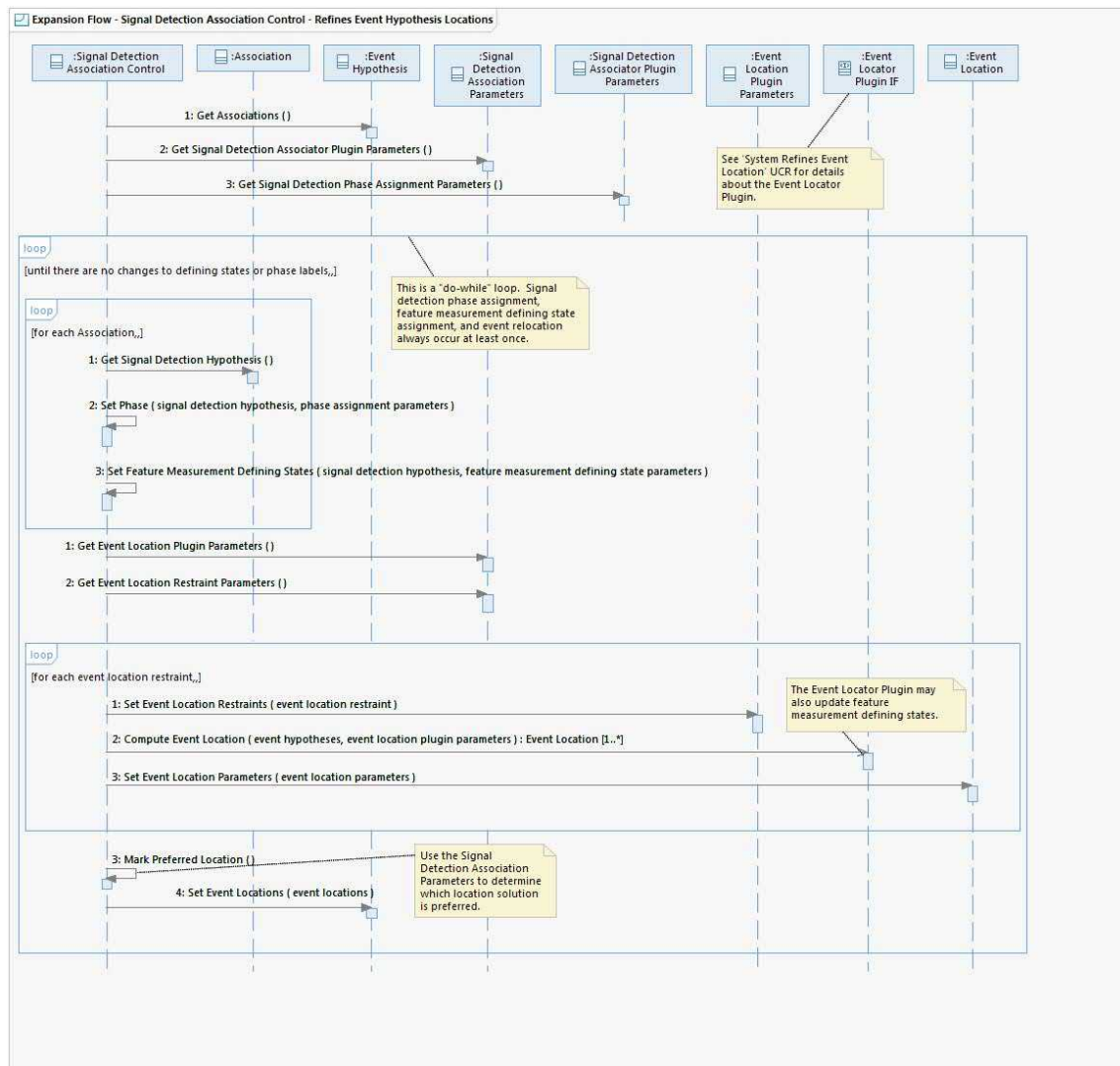
### ***Operation: Event Hypothesis::Create Copy()***

Create a copy of the given Event Hypothesis. The copy has all of the same information as the original (e.g. same detections, location, etc.), with the following exceptions:

- The copy points to the original as its parent
- The copy starts out with an empty Analyst comment



## Expansion Flow - Signal Detection Association Control - Refines Event Hypothesis Locations



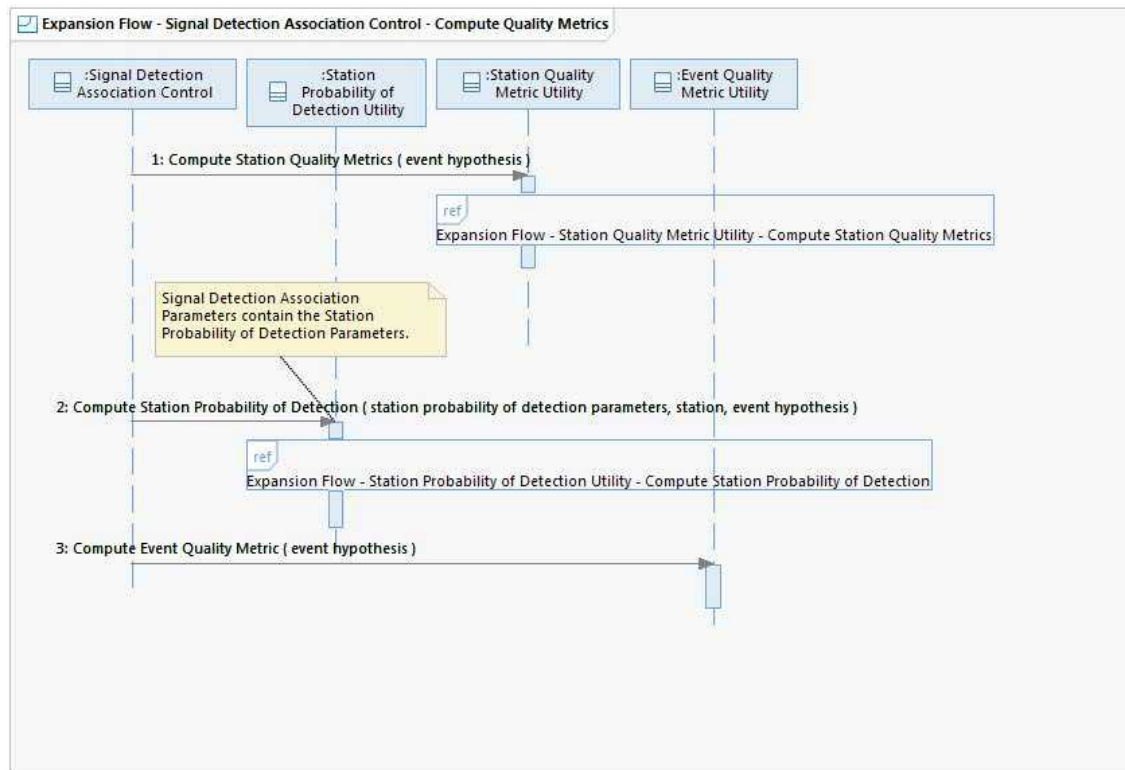
This flow shows Signal Detection Association Control iteratively setting signal detection phase labels and feature measurement defining states for all detections associated to an event hypothesis before refining the hypothesis' location. Iteration continues until relocating the hypothesis does not cause any changes to the phase labels or defining states.

### Operation Descriptions

#### **Operation: Event Locator Plugin IF::Compute Event Location()**

The interface method for an Event Location Plugin to compute an event location. The plugin may compute the location using teleseismic and regional seismic signal detections.

## Expansion Flow - Signal Detection Association Control - Compute Quality Metrics



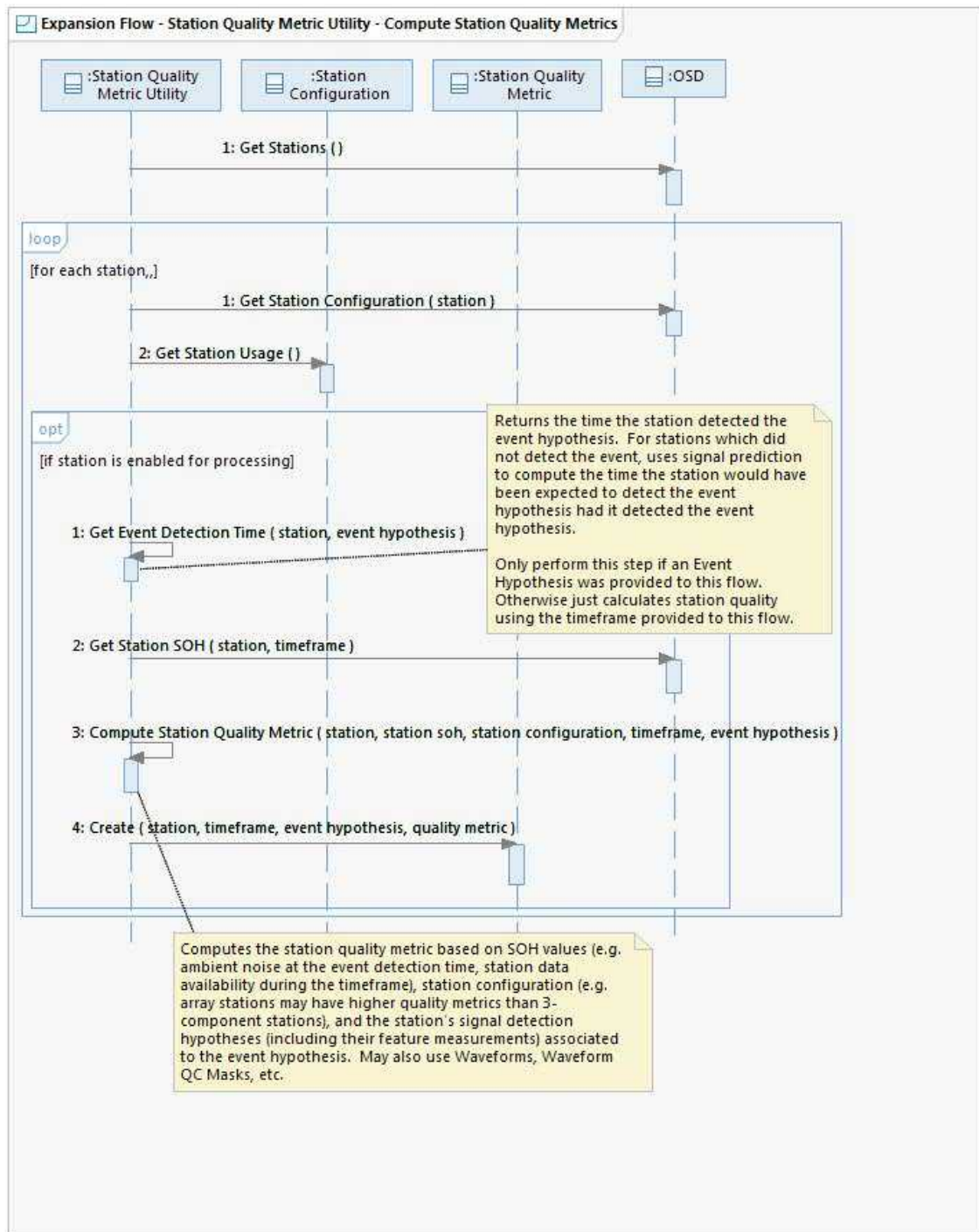
This flow shows Signal Detection Association Control computing station quality metrics, station probabilities of detection, and event quality metrics for an event hypothesis.

### Operation Descriptions

None



## Expansion Flow - Station Quality Metric Utility - Compute Station Quality Metrics



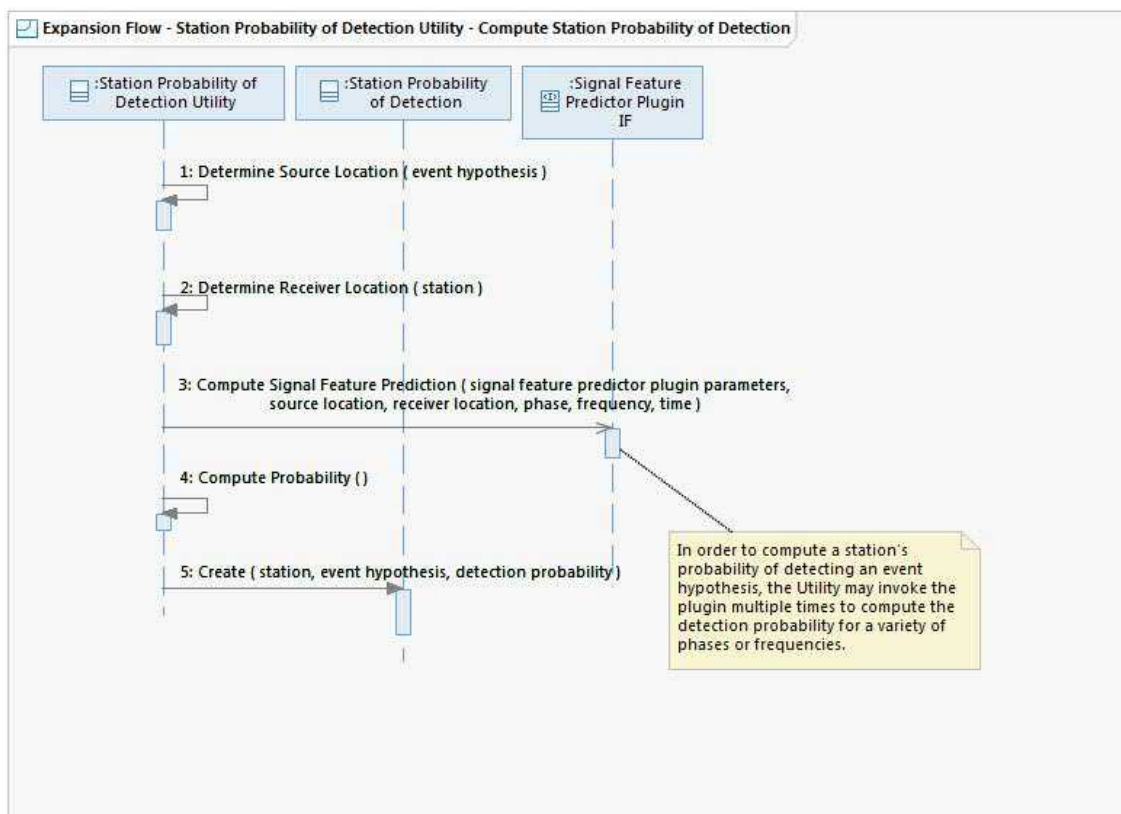
This flow shows Station Quality Metric Utility using information from Station Configuration, Station SOH, and Event Hypothesis to compute the station quality metric for each station. Station Quality Metric Utility computes quality metrics for the time each station detected the event hypothesis. For stations that did not detect the event hypothesis the utility computes the

quality metric using the station's predicted detection time for the hypothesis. Station Quality Metric Utility potentially computes multiple Station Quality Metric values for each Station (e.g. based on raw waveforms from the Station, based on a detection beam created by System Enhances Signals Control using the Station's waveforms).

## Operation Descriptions

None

### *Expansion Flow - Station Probability of Detection Utility - Compute Station Probability of Detection*

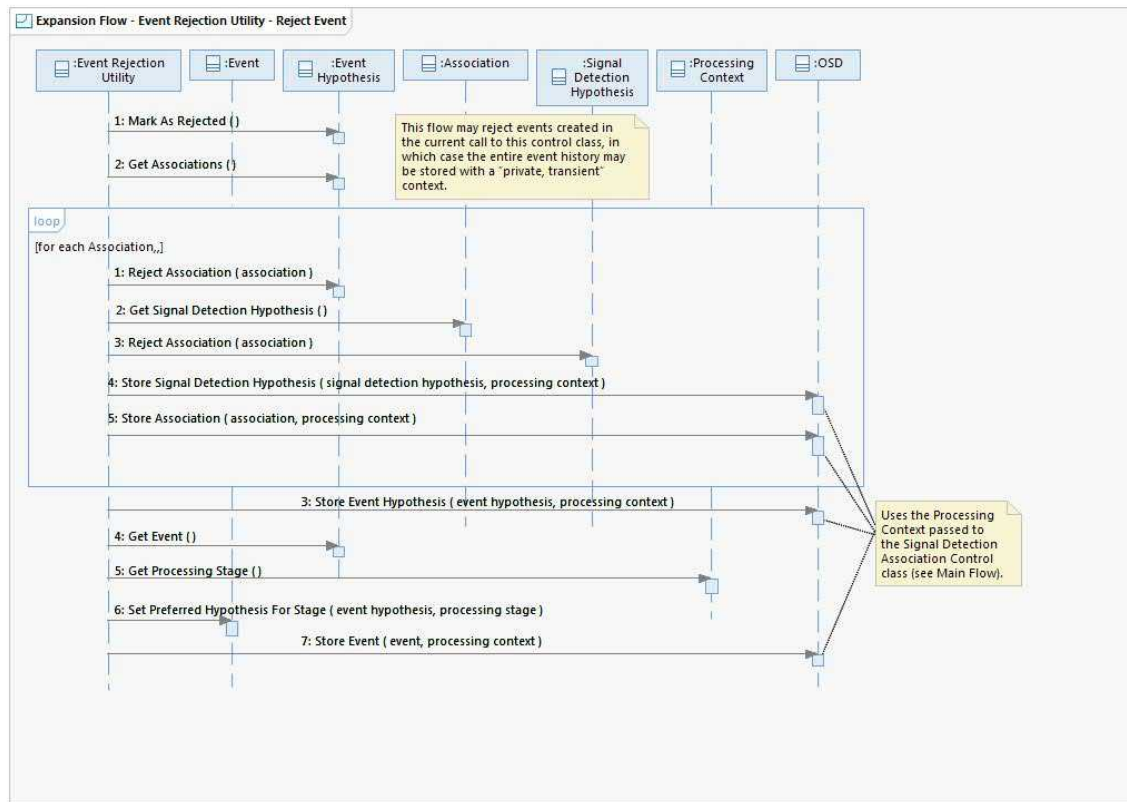


This flow shows the Station Probability of Detection Utility binding to and invoking a Signal Feature Predictor Plugin to calculate a Station's probability of detecting a provided Event Hypothesis. The invoking Control class determines the parameters defining the specific Signal Feature Predictor Plugin IF used by the Station Probably of Detection Utility as well as the phases and frequencies it uses in calls to the plugin. Station Probability of Detection Utility creates a Station Probability of Detection object to hold the results of this calculation.

## Operation Descriptions

None

## Expansion Flow - Event Rejection Utility - Reject Event



This flow shows Event Rejection Utility rejecting an event. The utility marks the provided event hypothesis as rejected, unassociates all signal detections from the event hypothesis, and sets this hypothesis as the event's preferred hypothesis. The utility stores the event, the event hypothesis, and the unassociated signal detection hypotheses in the OSD using the provided Processing Context.

### Operation Descriptions

#### **Operation: OSD::Store Event Hypothesis()**

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

#### **Operation: OSD::Store Event()**

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

### State Machine Diagrams

None

## SSD Mappings

### General:

**S-1504:** [*Threshold*] The System shall use the seismic, hydroacoustic, and infrasound network signal association parameters to form event hypotheses from any combination of seismic, hydroacoustic, and infrasound signal detections meeting the signal association event hypothesis formation criteria.

**S-1505:** [*Threshold*] The System shall use empirical values of the network signal association parameters during event hypothesis formation.

**S-1506:** [*Threshold*] The System shall use historic probability of signal detection during event hypothesis formation.

**S-1507:** [*Threshold*] The System shall use seismic network signal association parameters to assign phases to seismic signal detections.

**S-1508:** [*Threshold*] The System shall use hydroacoustic network signal association parameters to assign H phases to hydroacoustic signal detections at hydrophone stations.

**S-1509:** [*Threshold*] The System shall use hydroacoustic network signal association parameters to assign T phases to hydroacoustic signal detections from T-phase stations.

**S-1510:** [*Threshold*] The System shall use infrasound network signal association parameters to assign phases to infrasound signal detections.

**S-1511:** [*Threshold*] The System shall support concurrent signal association event hypothesis formation criteria.

**S-1513:** [*Threshold*] The System shall associate unassociated signal detections created by any signal detection algorithm to event hypotheses formed by any event formation algorithm.

**S-1514:** [*Threshold*] The System shall compute the station probability of detecting an event hypothesis during event formation.

**S-1515:** [*Threshold*] The System shall use variable resolution representations of the Earth for signal association parameter predictions during signal association to account for the varying ability to resolve signals originating in different areas.

**S-1516:** [*Objective / Priority 2*] The System shall create new event hypotheses which modify existing user-reviewed event hypotheses only when the event quality metric for the automatic event hypothesis improves more than a configured threshold.

**S-1518:** [*Threshold*] The System shall use the configured earth model(s) during signal detection association.

**S-1540:** [*Threshold*] The System shall perform late network signal association using the seismic, hydroacoustic, and infrasound network signal association parameters to form event hypotheses from any combination of seismic, hydroacoustic, and infrasound signal detections meeting the signal association event hypothesis formation criteria.

**S-1541:** [*Threshold*] The System shall perform late network signal association for a time interval when one or more signal detections for that time interval become available that were not available during prior network signal association or late network signal associations for that time interval.

**S-1542:** [*Threshold*] The System shall not automatically perform network signal association affecting signal detections that the Analyst is actively reviewing.

**S-1543:** [*Threshold*] The System shall set signal detections to non-defining for event hypothesis location calculations when the System automatically associates them to Analyst reviewed event hypotheses.

**S-1544:** [*Threshold*] The System shall set station magnitudes to non-defining for event hypothesis magnitude calculations when the System automatically associates them to Analyst reviewed event hypotheses.

**S-1548:** [*Threshold*] The System shall not automatically reassociate a signal detection to an event hypothesis if an Analyst has previously unassociated that signal detection from the event hypothesis.

**S-1554:** [*Threshold*] The System shall set to non-defining newly associated signal detections when the Analyst invokes automated processing algorithms to associate signal detections to existing event hypotheses.

**S-1556:** [*Threshold*] The System shall store all event hypotheses formed by the System.

**S-1557:** [*Threshold*] The System shall store all signal detection associations for each event hypothesis stored by the System.

**S-1560:** [*Threshold*] The System shall associate signal detections to event hypotheses found with waveform correlation event processing by mapping signal detections on all network sensors to compatible signal detections on the historic waveforms.

**S-1561:** [*Threshold*] The System shall assign phase identifications to signal detections associated to event hypotheses found via waveform correlation to the same phases as the matching historic event hypotheses.

- S-1572:** [*Threshold*] The System shall compute the station quality metric for all events.
- S-1576:** [*Threshold*] The System shall store the station quality metrics for all stations for each event hypothesis.
- S-1579:** [*Threshold*] The System shall compute an event hypothesis quality metric using the event hypothesis quality statistics for each event hypothesis formed on the System.
- S-1580:** [*Threshold*] The System shall recompute the event hypothesis quality metric for an event hypothesis when any of the event hypothesis quality statistics used to calculate the event hypothesis quality metric are updated.
- S-1581:** [*Threshold*] The System shall not automatically form event hypotheses with event hypothesis quality metrics below the event hypothesis quality metric threshold.
- S-1582:** [*Threshold*] The System shall not screen any Analyst created event hypotheses by their event hypothesis quality metrics.
- S-1588:** [*Threshold*] The System shall store the event quality metric for each event hypothesis.
- S-1597:** [*Threshold*] The System shall compute new event hypothesis relocations when an automatic process associates a new location defining signal detection to that event hypothesis.
- S-1598:** [*Threshold*] The System shall compute new event hypothesis relocations when an automatic process unassociates a location defining signal detection from that event hypothesis.
- S-2036:** [*Threshold*] The System shall use configured default defining/non-defining state settings and precedence rules to determine the initial defining/non-defining state for each parameter.
- S-2043:** [*Threshold*] The System shall store automatic and interactive processing results.
- S-2223:** [*Threshold*] The System shall store all data and derived processing results to persistent storage as soon as the data and/or derived processing results are available.
- S-2342:** [*Threshold*] The System shall mark as requiring Analyst review any event previously reviewed by an Analyst that has its signal detection associations subsequently modified by the System.
- S-3026:** [*Threshold*] The System shall build new events using signal detection templates.
- S-5596:** [*Threshold*] The System shall use station-to-event distance when associating signal detections to events.
- S-5597:** [*Threshold*] The System shall use event magnitude when associating signal detections to events.

**S-5598:** [*Threshold*] The System shall use waveform data quality when associating signal detections to events.

**S-5599:** [*Threshold*] The System shall use station noise level when associating signal detections to events.

**S-5600:** [*Threshold*] The System shall use event location to assign phase identifications to signal detections based on predicted phase ID matching.

**S-5601:** [*Threshold*] The System shall use event location to assign phase identifications to signal detections based on available empirical phase ID matching.

**S-5641:** [*Objective / Priority 1*] The System shall identify microbarom signals.

**S-5642:** [*Objective / Priority 1*] The System shall prevent association of microbarom signals to events.

**S-5968:** [*Threshold*] The System shall associate signal detections to existing events using signal detection templates.

## Notes

### General:

1. “Expansion Flow – Signal Detection Association Control - Associate Signal Detections” shows Signal Detection Association Control calling a single Signal Detection Associator Plugin. The System Maintainer may configure a Processing Sequence to call multiple Signal Detection Associator Plugins by configuring the Processing Sequence to invoke Signal Detection Association Control more than one time but with each call using parameters specifying a different plugin (see ‘Defines Processing Sequence’ UCR).
2. In “Expansion Flow – Signal Detection Association Control – Associate Signal Detections” Signal Detection Association Control creates a new Event for each new Event Hypothesis built by a Signal Detection Associator Plugin. Signal Detection Association Control may reject some of these Events if they have poor event quality metrics (see “Expansion Flow – Signal Detection Association Control – Postprocess Event Hypotheses”) and other Events may be modified or rejected during conflict resolution (see ‘System Resolves Event Conflicts’ UCR). As an optimization, the implementation may decide to not create Event objects for newly built Event Hypotheses until after conflict resolution.
3. ‘System Resolves Event Conflicts’ UCR describes how signal detection association conflicts are resolved, including conflict checking between event hypotheses built in this UCR and Analyst reviewed event hypotheses.

**IDC Specific:**  
None.



# IDC Use Case Realization Report

## **UCR-03.03 Scans Waveforms and Unassociated Detections**

### **Use Case Description**

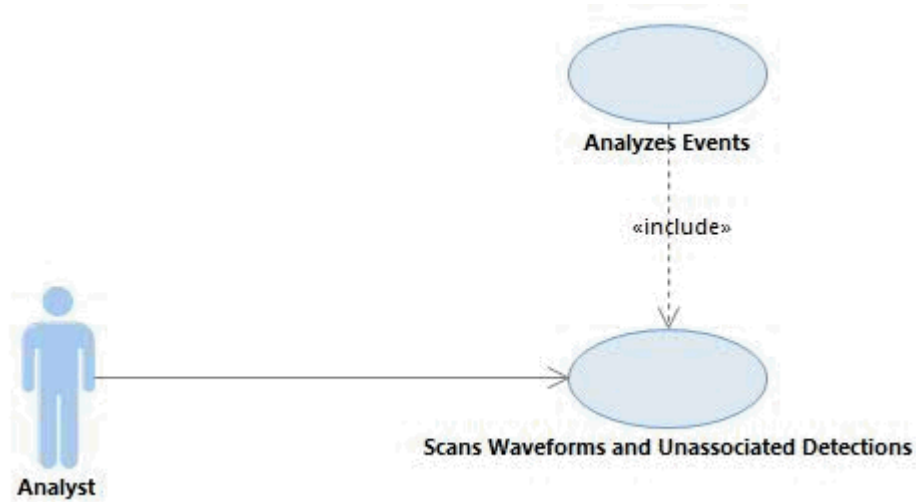
This architecturally significant use case describes how the Analyst scans waveforms or detection feature maps and unassociated signal detections to build new events. The Analyst scans unassociated signal detections to find signal detections that should be associated to existing event hypotheses, and scans waveforms and detection feature maps to find signal detections missed by prior processing. The Analyst uses two data scanning methods. The Analyst scans waveforms through time to detect new signals and associate unassociated signal detections to existing event hypotheses (see ‘Detects Signals’ UC).

This use case is architecturally significant because it provides a platform for the Analyst to efficiently review large amounts of sensor data in order to evaluate, correct and improve signal detection and event formation.

### **Architecture Description**

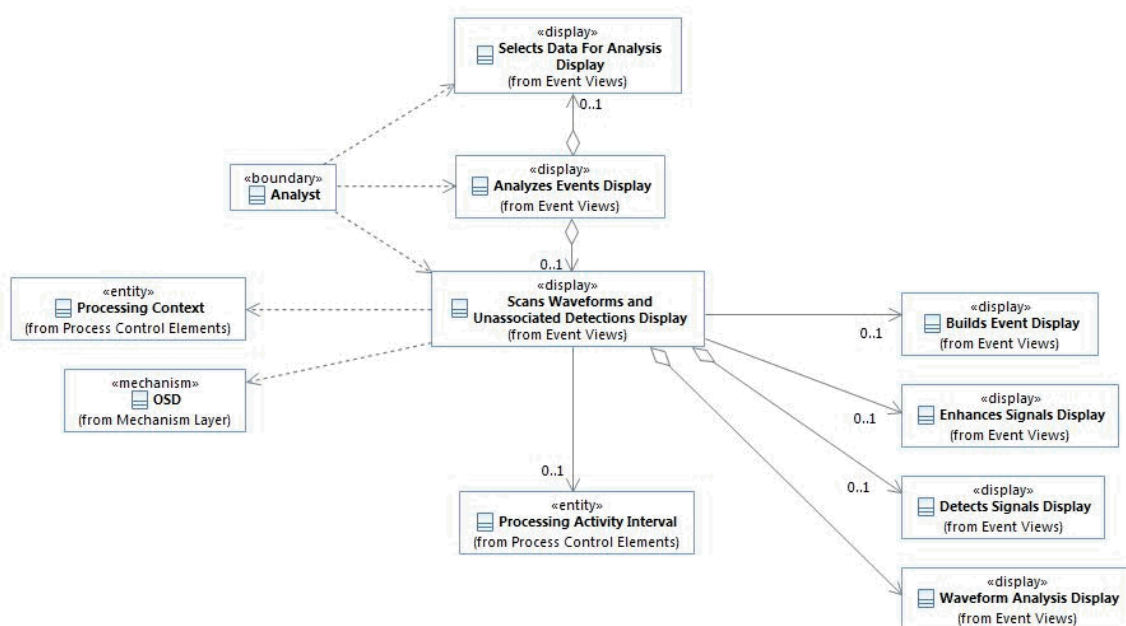
The Analyst begins scanning Waveforms, Detection Feature Maps, and unassociated signal detections by selecting a scanning activity from the Selects Data for Analysis Display. The Analyst uses the Scans Waveforms and Unassociated Detections Display to view and scan through the Waveforms, Detection Feature Maps, and unassociated signal detections. The Analyst uses the Enhances Signals Display to filter waveforms to emphasize signals. The Analyst uses the Detects Signals Display to create signal detections, refine signal detections, reject signal detections, associate signal detections to event hypotheses, and to modify existing signal detection associations to event hypotheses. The Analyst uses the Builds Events Display to create new event hypotheses. These actions are depicted in those use case realizations. As the Analyst creates or modifies Signal Detections, Event Hypotheses, or Events the objects are stored transiently in a private context via the OSD mechanism to make them available to the Processing Sequence Control mechanism for further automatic processing (the Processing Sequence Control mechanism is described in ‘System Detects Event’ UCR). When the Analyst selects to save changes the Scans Waveforms and Unassociated Detections Display uses the OSD to store the Events, Event Hypotheses, unassociated Signal Detections, Waveforms, and Detection Feature Maps with a persistent, global context.

## Use Case Diagram



## Class Diagrams

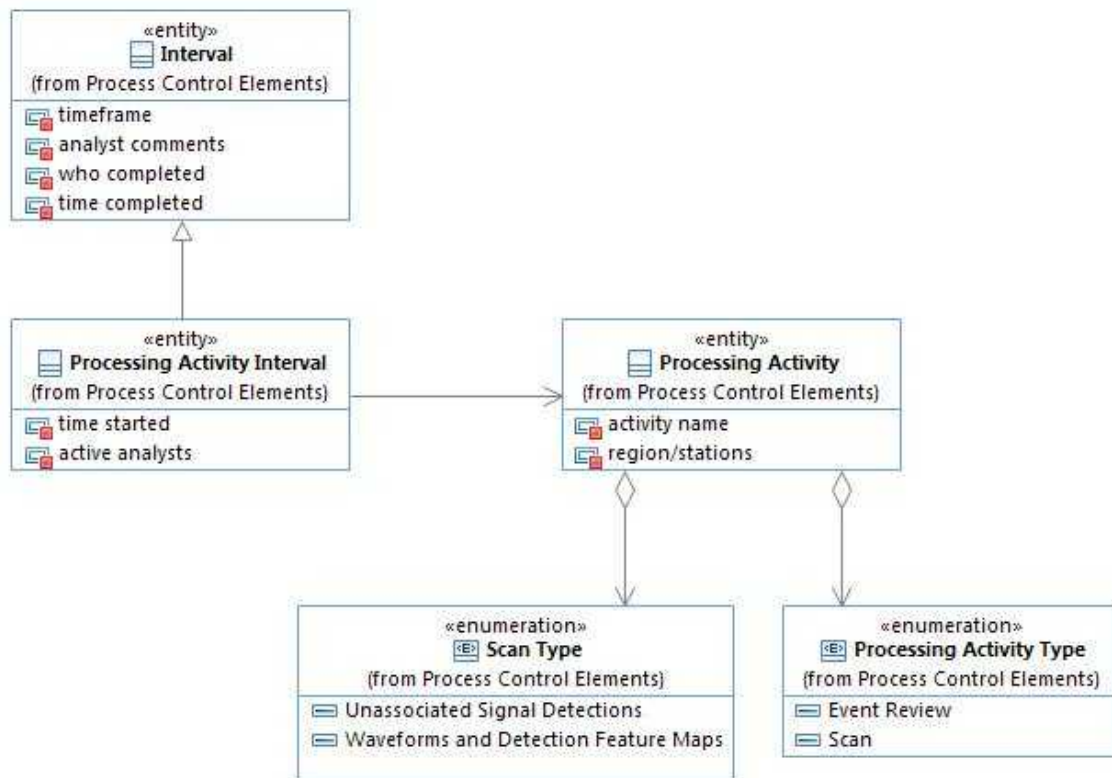
### *Classes - Scans Waveforms and Unassociated Detections Display*



This diagram shows the Scans Waveforms and Unassociated Detections Display and its relationships to other classes as relevant to this use case. The display is created by the Analyzes Events Display when the Analyst selects to perform a scanning Processing Activity. The display uses the Waveform Analysis Display to show Waveforms, Detection Feature Maps indexed to the Waveforms used to build them, and Signal Detections. The Waveform Analysis Display provides the Analyst with the capability to scan the Detection Feature Maps and to create, reject, and modify signal detections. The Scans Waveforms and Unassociated Detections Display shows

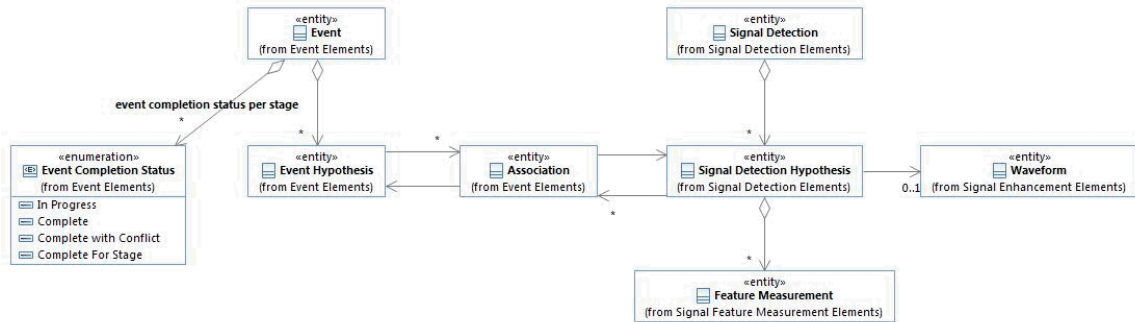
Events and allows the Analyst to scroll through the data. The display uses the Builds Events Display to create Events (and Event Hypotheses), uses the Enhances Signals Display to create derived waveforms and Detection Feature Maps, uses the Detects Signals Display to create Signal Detections (and Signal Detection Hypotheses), and uses the Detects Signals Display to associate Signal Detection Hypotheses to Event Hypotheses. New or modified unassociated Signal Detections (and Signal Detection Hypotheses), and new or modified Events (and Event Hypotheses) are stored in the OSD. The OSD also provides data callbacks when other Analysts or the System update data related to the scan.

### ***Classes - Processing Activity Interval***



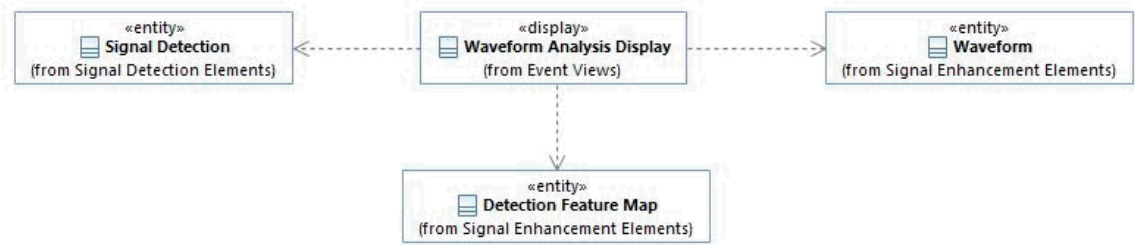
This diagram shows the Processing Activity Interval class and its relationships when used to define a scanning Processing Activity.

## Classes - Event



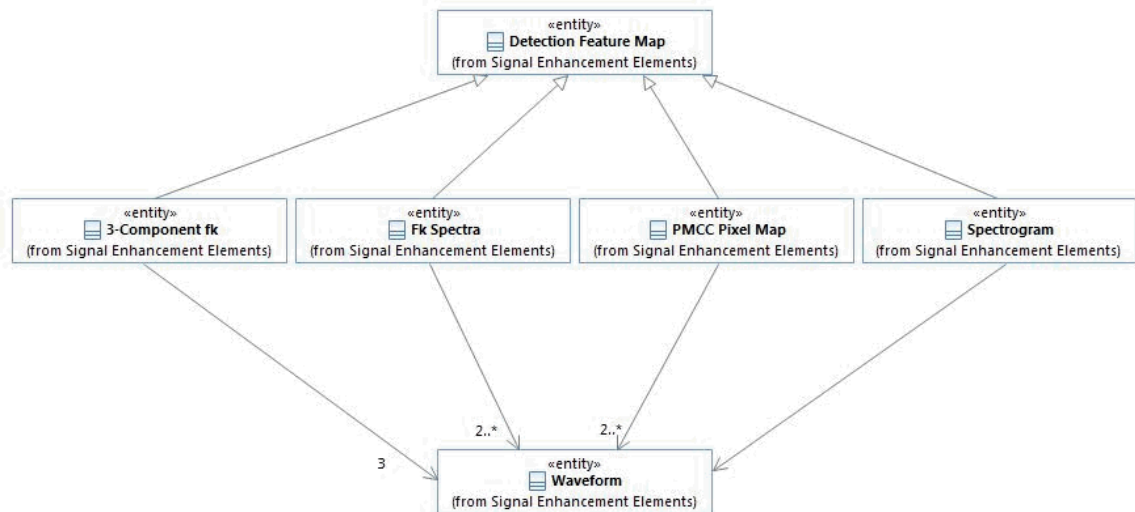
This class diagram shows the Event class and its associated classes as used in ‘Scans Waveforms and Unassociated Detections’ UCR. While scanning waveforms and unassociated detections the Analyst creates new Signal Detections, modifies existing Signal Detection Hypotheses, rejects existing Signal Detections, builds new Event Hypotheses, and associates Signal Detection Hypotheses to Event Hypotheses. While creating Signal Detection Hypotheses the Analyst may create new Waveforms by enhancing existing Waveforms. These actions are depicted in ‘Enhances Signals’ UCR, ‘Detects Signals’ UCR, and ‘Builds Event’ UCR. All Events created or modified in this UCR are stored with an Event Completion Status of “In Progress”. Further refinement and setting the Event Completion Status to other values is shown in ‘Refines Event’ UCR and ‘Marks Processing Stage Complete’ UCR.

## Classes - Waveform Analysis Display



This diagram shows the relationship of the Waveform Analysis Display with the Detection Feature Maps, signal detections, and waveforms.

## Classes - Detection Feature Map



This diagram shows the relationship of the Detection Feature Map with Waveform and the classes that specialize Detection Feature Map.

## Class Descriptions

### **<<boundary>> Analyst**

Represents the Analyst actor.

### **<<display>> Analyzes Events Display**

Display that provides the Analyst with the ability to analyze data within a specified time interval in order to find or refine Events.

### **<<display>> Detects Signals Display**

Provides the Analyst with the ability to rerun automatic signal detection with Analyst-specified parameters.

### **<<display>> Scans Waveforms and Unassociated Detections Display**

The Scans Waveforms and Unassociated Detections Display shows waveforms (this includes raw waveforms, filtered waveforms, origin beams, virtual origin beams, fk beams, and other derived waveforms), Events, Signal Detection Hypotheses associated to those Events, and unassociated Signal Detection Hypotheses. The Scans Waveforms and Unassociated Detections Display has a different look and feel depending on the selected scan type. For example, which station's waveforms are displayed and waveform sorting may change based on the scan type.

### **<<display>> Selects Data For Analysis Display**

Display that provides the Analyst with the ability to select an interval or event set to analyze. Displays the current workflow status for all intervals as well as events marked by the system or an analyst as events of interest.

**<<display>> *Waveform Analysis Display***

Displays a set of waveforms and Detection Feature Maps and provides the Analyst with the ability to interact with them (e.g. create/modify/reject Signal Detections, associate/unassociate detections and Events).

**<<entity>> *3-Component fk***

Represents a 3-Component fk. This is an approximation of an fk spectra calculated for a 3-Component station by performing polarization analysis for the station's waveforms rotated to a variety of points in azimuth-slowness space.

**<<entity>> *Association***

Represents an association between a Signal Detection Hypothesis and an Event Hypothesis.

**<<entity>> *Detection Feature Map***

A Detection Feature Map is a base class representing multi-dimensional features computed from a Station's waveforms over time. Specializations of Detection Feature Map determine the measured feature (e.g. Spectrogram measures signal amplitude indexed by frequency).

**<<entity>> *Event***

Represents information about an Event. Keeps track of all the Event Hypotheses for the Event, which Event Hypothesis is the preferred one for each processing stage, the active analysts for the Event (i.e. whether the Event is under "active review"), whether the Event is "complete" for each processing stage, and other Event-related information.

**<<entity>> *Event Hypothesis***

Represents geophysical information about an Event as determined by an Analyst or through pipeline processing. There can be multiple Event Hypotheses for the same Event (e.g. different associated Signal Detection Hypotheses, different location solutions).

**<<entity>> *Feature Measurement***

Represents the value and uncertainty of a measured feature of a signal detection.

**<<entity>> *Fk Spectra***

Represents the fk (frequency-wavenumber) power spectra and associated F-statistics for a set of waveforms. The fk power spectrum is a representation in the fk domain of coherent signal power in waveform data from multiple stations for a specified time interval. Calculation of fk power spectrum requires three Fourier transforms to convert time-sampling to frequency (designated by f), and longitude and latitude sampling to x and y wavenumbers (designated by k). Typically fk power spectra are represented as 2D plots (not 3D), by collapsing the frequency information to a single value for each x and y slowness by averaging values across the range of frequencies.

**<<entity>> *Interval***

Class for tracking the status of interactive or automatic processing on a specific timeframe of data. Specialized intervals exist for Processing Stage, Processing Activity, and Processing

Sequence.

**<<entity>> *PMCC Pixel Map***

Represents a PMCC Pixel Map which contains pixels of coherent waveform segments from multiple sensors prior to grouping the pixels into families. Each pixel may contain measurements such as horizontal trace velocity and back azimuth.

**<<entity>> *Processing Activity***

Defines an activity within a processing stage. Two types of Processing Activities exist: Event Review and Scan. The "region/stations" attribute may be set by the user to optionally limit the activity to the specified geographical region or set of stations.

**<<entity>> *Processing Context***

Represents the context in which data is being stored and/or processed. This includes the Processing Stage (either automatic or interactive) and Interval performing the processing session (e.g. processed by Analyst vs. processed by System). For Analyst processing, may identify the Analyst work session. For System processing, may identify the Processing Sequence and/or Processing Step being executed (including a way to identify a particular Processing Sequence and Processing Step among the many possible instantiations), the visibility for the results (private vs. global), and the lifespan of the data (transient vs. persistent). This information is needed by the Processing Sequence Control to manage the execution of Processing Sequences, which may execute in the context of an Analyst refining an Event or in the context of the system initiating automatic processing. It is also needed by the Object Storage and Distribution (OSD) mechanism to determine how to store and distribute the data.

**<<entity>> *Signal Detection***

Represents information about a Signal Detection and keeps track of all the Signal Detection Hypotheses for the Signal Detection. Represents information about a Signal Detection and keeps track of all the Signal Detection Hypotheses for the Signal Detection. For an unassociated Signal Detection the preferred Signal Detection Hypothesis is the most recently created Signal Detection Hypothesis. For an associated Signal Detection the preferred Signal Detection Hypothesis is the one associated to a preferred Event Hypothesis.

**<<entity>> *Signal Detection Hypothesis***

Represents geophysical information about a Signal Detection as determined by an Analyst or through pipeline processing. There can be multiple Signal Detection Hypotheses for the same Signal Detection (e.g. different onset times, different phase labels).

**<<entity>> *Spectrogram***

Indicates how the frequency content of the associated Waveform varies over time.

**<<entity>> *Waveform***

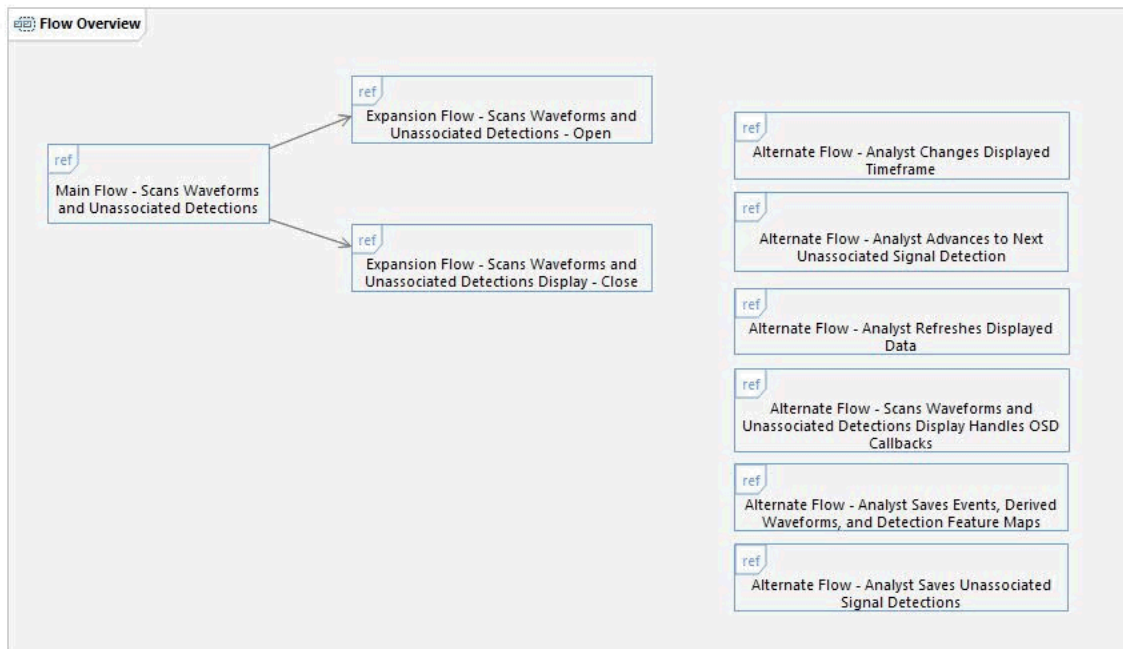
A Waveform represents a time-series of data from a Channel. The Channel can be a Raw Channel, representing data from a Station, or a Derived Channel, representing data after a signal enhancement has been applied.

## <<mechanism>> OSD

Represents the Object Storage and Distribution mechanism for storing and distributing data objects internally within the system.

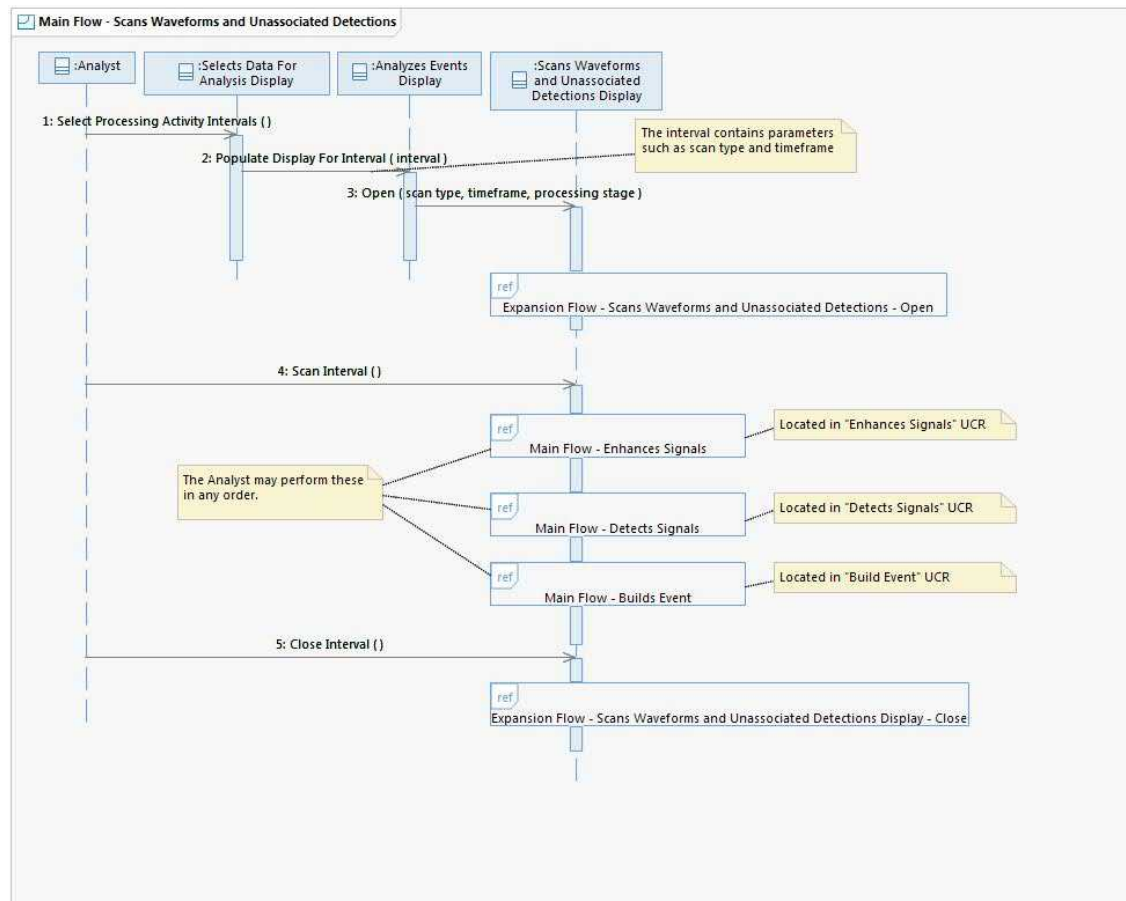
## Sequence Diagrams

### Flow Overview





## Main Flow - Scans Waveforms and Unassociated Detections

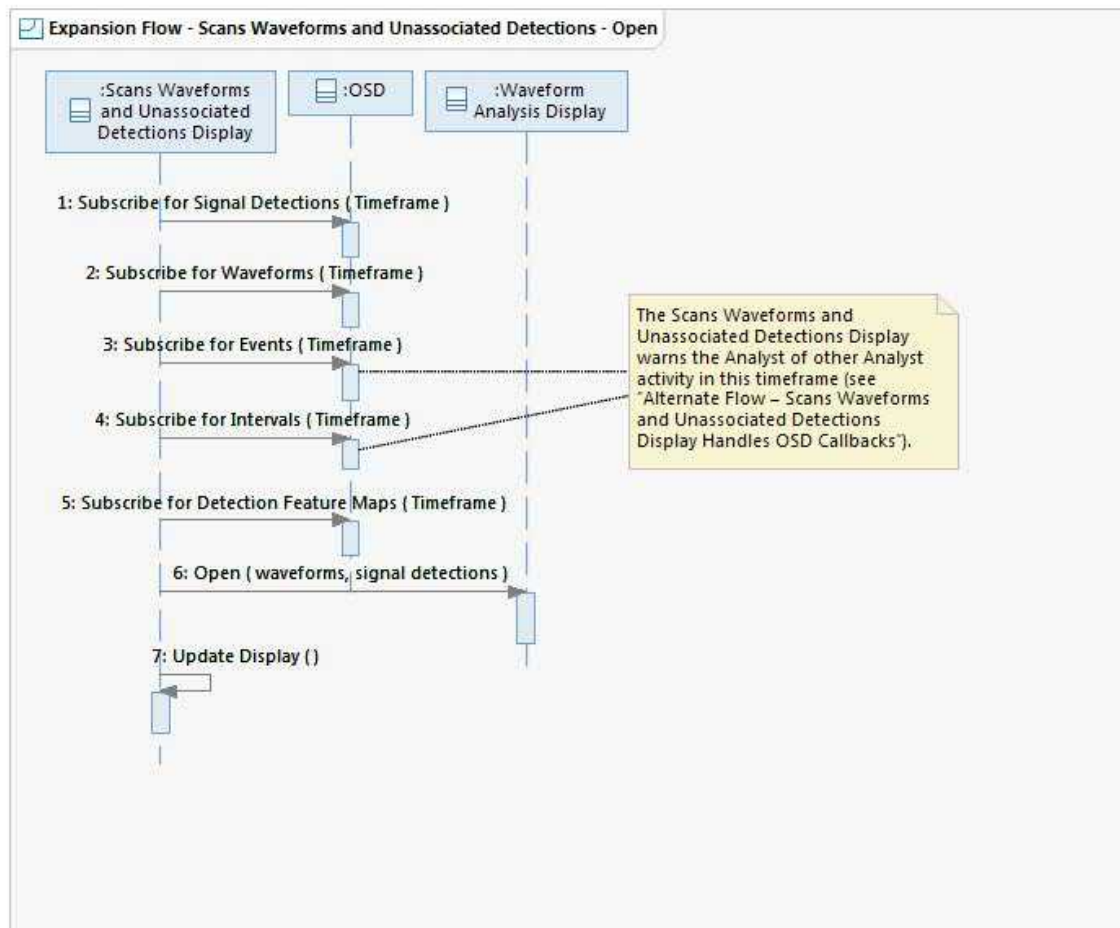


This flow shows the Analyst's primary actions when scanning waveforms and unassociated detections. When the Analyst selects to process a Processing Activity Interval as described in 'Selects Data for Analysis' UCR the scan type is the type associated with the selected Interval, and the timeframe parameter is the timeframe corresponding to the Interval the Analyst previously selected for analysis, and the processing stage is the same as that Interval's processing stage. The Analyst scans waveforms, Detection Feature Maps, and unassociated detections by enhancing signals (see 'Enhances Signals' UCR), creating, updating, and/or rejecting Signal Detections (see 'Detects Signals' UCR), and building and/or updating Events (see 'Builds Events' UCR). The GUI in the Scans Waveforms and Unassociated Detections Display is opened with a different visual layout that is based on the scan type. When the scan is complete, the Analyst closes the Scans Waveforms and Unassociated Detections Display.

### Operation Descriptions

None

## Expansion Flow - Scans Waveforms and Unassociated Detections - Open



This flow shows opening the Scans Waveforms and Unassociated Detections Display for a timeframe and scan type. The Scans Waveforms and Unassociated Detections Display registers with the OSD for data updates. The subscription Timeframes correspond to the timeframe parameter passed to this flow, which spans the entire timeframe of data to be scanned. The Scans Waveforms and Unassociated Detections Display populates the display.

### Operation Descriptions

#### **Operation: *OSD::Subscribe for Signal Detections()***

Subscribe for updates regarding Signal Detection creations, modifications, and associations occurring within the specified timeframe. This includes updates for new or modified unassociated Signal Detections.

#### **Operation: *OSD::Subscribe for Waveforms()***

Subscribe for updates regarding raw and derived waveforms occurring within a specified timeframe. This includes information about what waveforms have been acquired by the System as well as what derived waveforms have been formed, but does not include the actual waveform data.

**Operation: *OSD::Subscribe for Events()***

Subscribe for changes to Event objects within the given timeframe. Callbacks are invoked on subscribers any time an Event within the timeframe is added or modified.

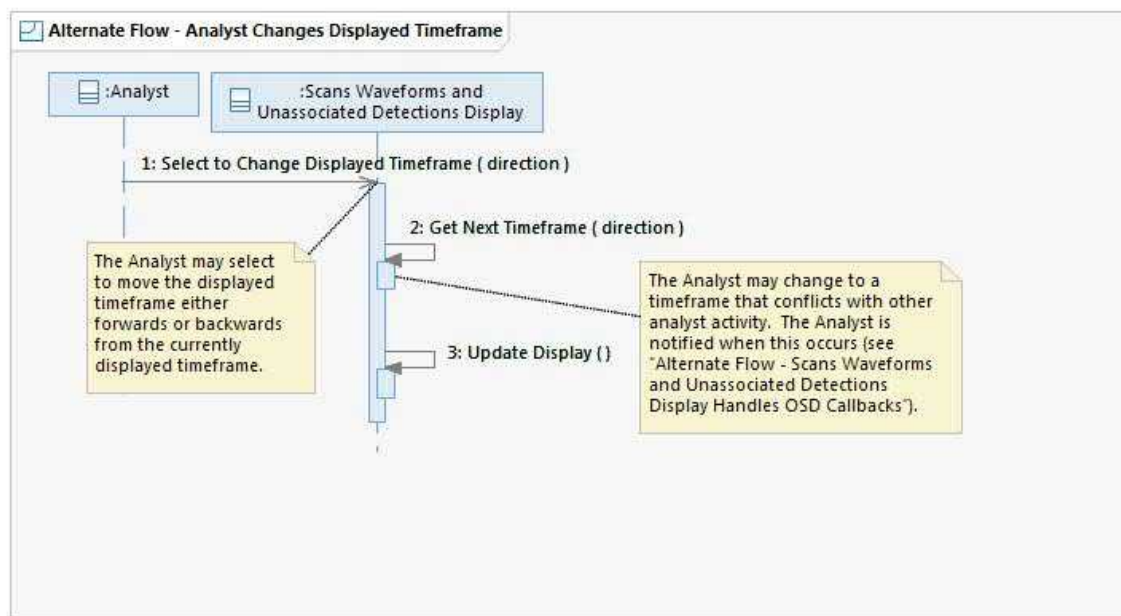
**Operation: *OSD::Subscribe for Intervals()***

Subscribe for changes to Interval objects that overlap with the given timeframe. Interval objects track the active analysts and completion status of intervals corresponding to processing stages and processing activities within processing stages. Callbacks are invoked on subscribers any time the set of active analysts or completion status for an Interval changes.

**Operation: *OSD::Subscribe for Detection Feature Maps()***

Subscribe for updates regarding Detection Feature Map creations and modifications within the specified timeframe.

***Alternate Flow - Analyst Changes Displayed Timeframe***

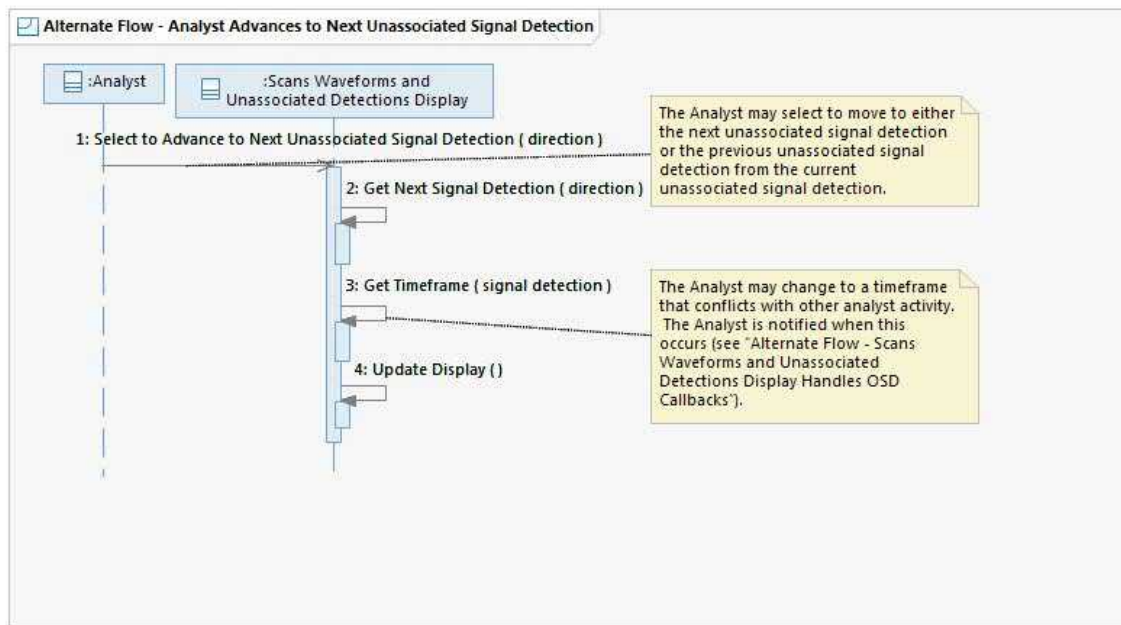


This flow shows how the Analyst changes the timeframe of waveforms and Detection Feature Maps visible in the Scans Waveforms and Unassociated Detections Display. The Analyst selects to scroll the timeframe to show waveforms either before or after the currently displayed waveforms.

**Operation Descriptions**

None

### ***Alternate Flow - Analyst Advances to Next Unassociated Signal Detection***



This flow shows how the Analyst changes the selected unassociated Signal Detection on the Scans Waveforms and Unassociated Detections Display. The Analyst selects to change to either the previous or the next unassociated signal detection. The Scans Waveforms and Unassociated Detections Display determines the new active timeframe as a timeframe centered on the selected Signal Detection Hypothesis.

#### **Operation Descriptions**

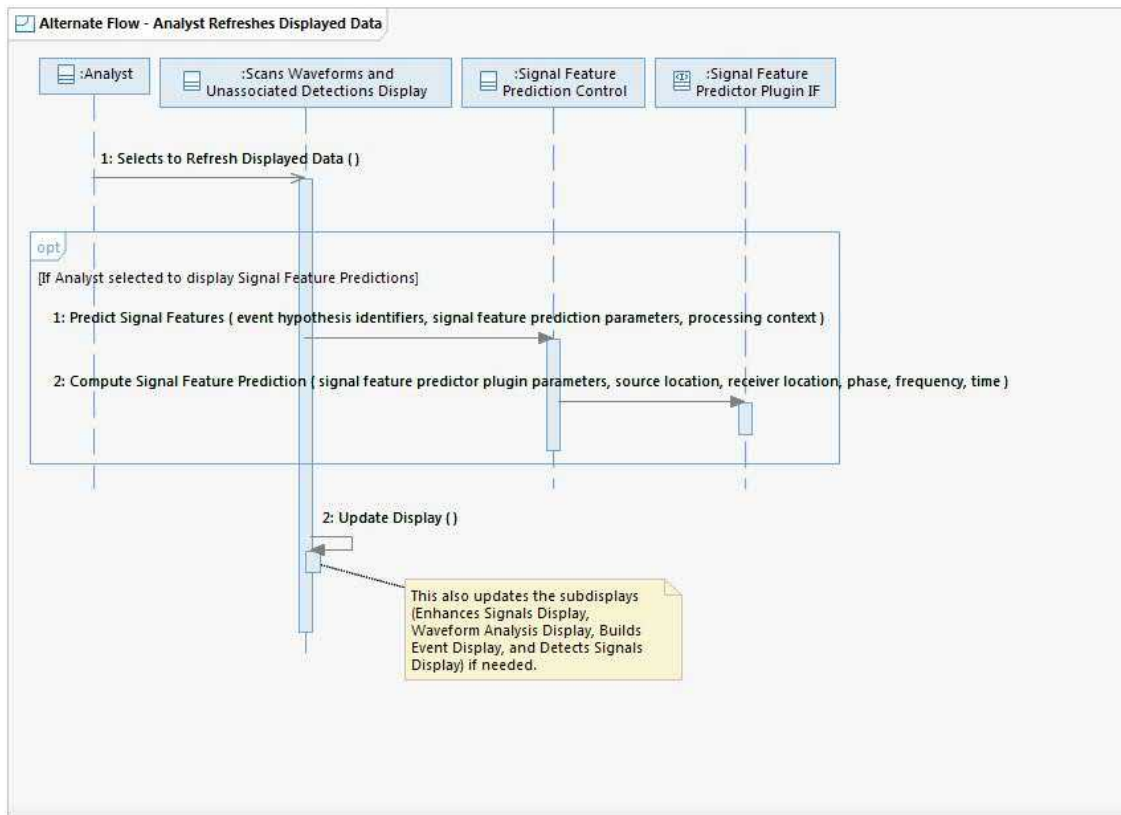
##### ***Operation: Scans Waveforms and Unassociated Detections Display::Get Timeframe()***

Computes an active timeframe that is centered on the provided signal detection.

##### ***Operation: Scans Waveforms and Unassociated Detections Display::Select to Advance to Next Unassociated Signal Detection()***

Based on the direction parameter, the Scans Waveforms and Unassociated Detections Display advances the selected Signal Detection to either the previous or the next unassociated Signal Detection.

### *Alternate Flow - Analyst Refreshes Displayed Data*

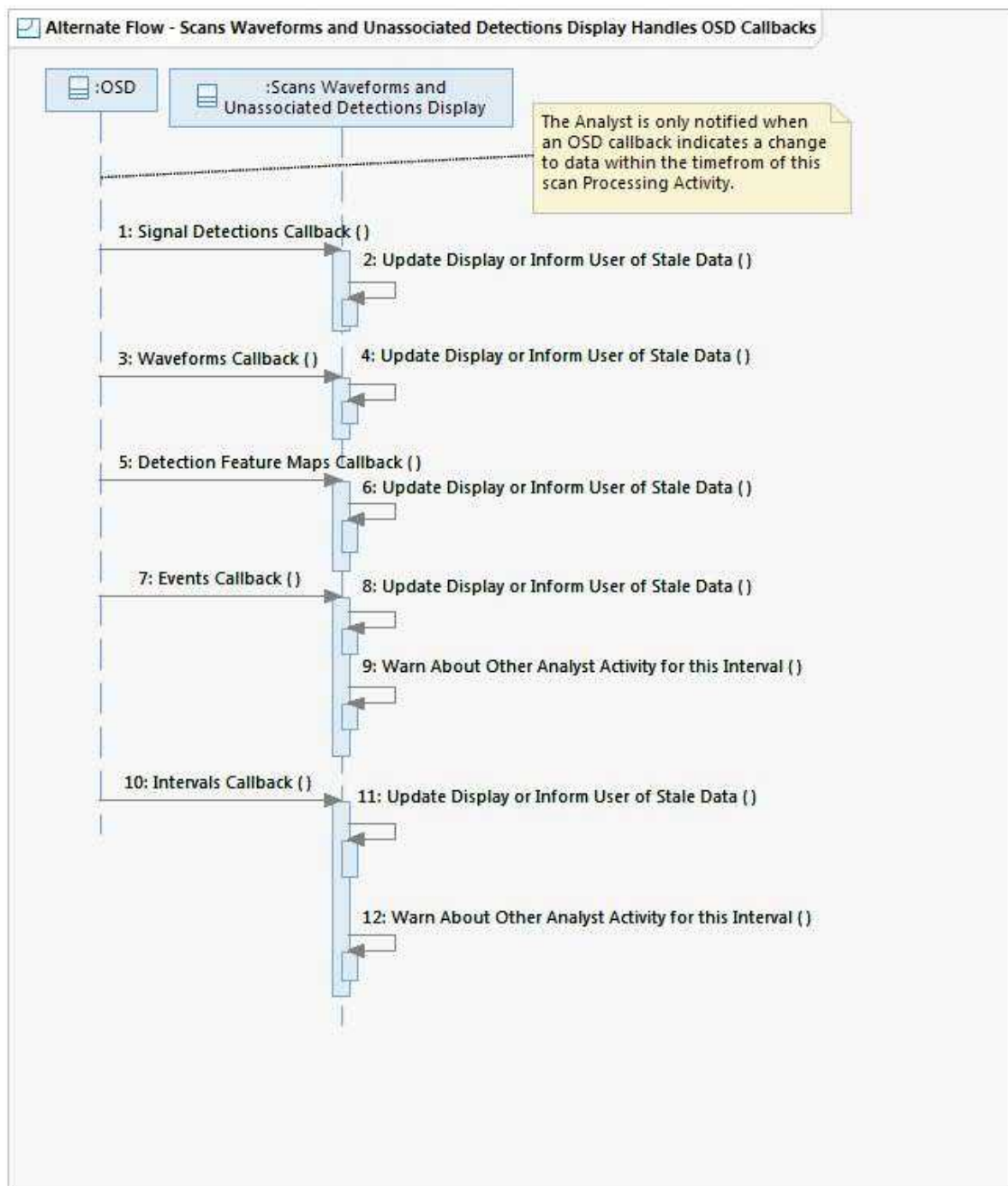


This flow shows how the Analyst refreshes the Scans Waveforms and Unassociated Detections Display to show the latest Waveforms, Detection Feature Maps, Predicted Signal Features, Signal Detections, and Events. The Analyst needs this capability since newly acquired Waveforms, Detection Feature Maps, new or modified Signal Detections from other Processing Stages, and new or modified Events from other Processing Stages are only updated when requested by the Analyst. The Analyst may also select to display predicted signal features such as predicted signal detection arrival times. Scans Waveforms and Unassociated Detections Display notifies the Analyst when there is new data that is not shown on the display (see "Alternate Flow - Scans Waveforms and Unassociated Detections Display Handles OSD Callbacks"). The Analyst may refresh the display to show that data at any time via this flow. Note that the display does not need to retrieve the new data from the OSD since it already has it due to subscriptions with the OSD (see "Expansion Flow - Scans Waveforms and Unassociated Detections – Open").

### **Operation Descriptions**

None

### ***Alternate Flow - Scans Waveforms and Unassociated Detections Display Handles OSD Callbacks***



This flow shows how the Scans Waveforms and Unassociated Detections Display handles callbacks from the OSD. The OSD invokes these callbacks as appropriate when data is stored in the OSD, which may be initiated by the current Analyst, other Analysts, or by the System as part of executing a Processing Sequence (executing Processing Sequences is shown in 'System Detects Event' UCR.) The Scans Waveforms and Unassociated Detections Display subscribes for Intervals and Events in order to notify the Analyst when other Analysts may be performing

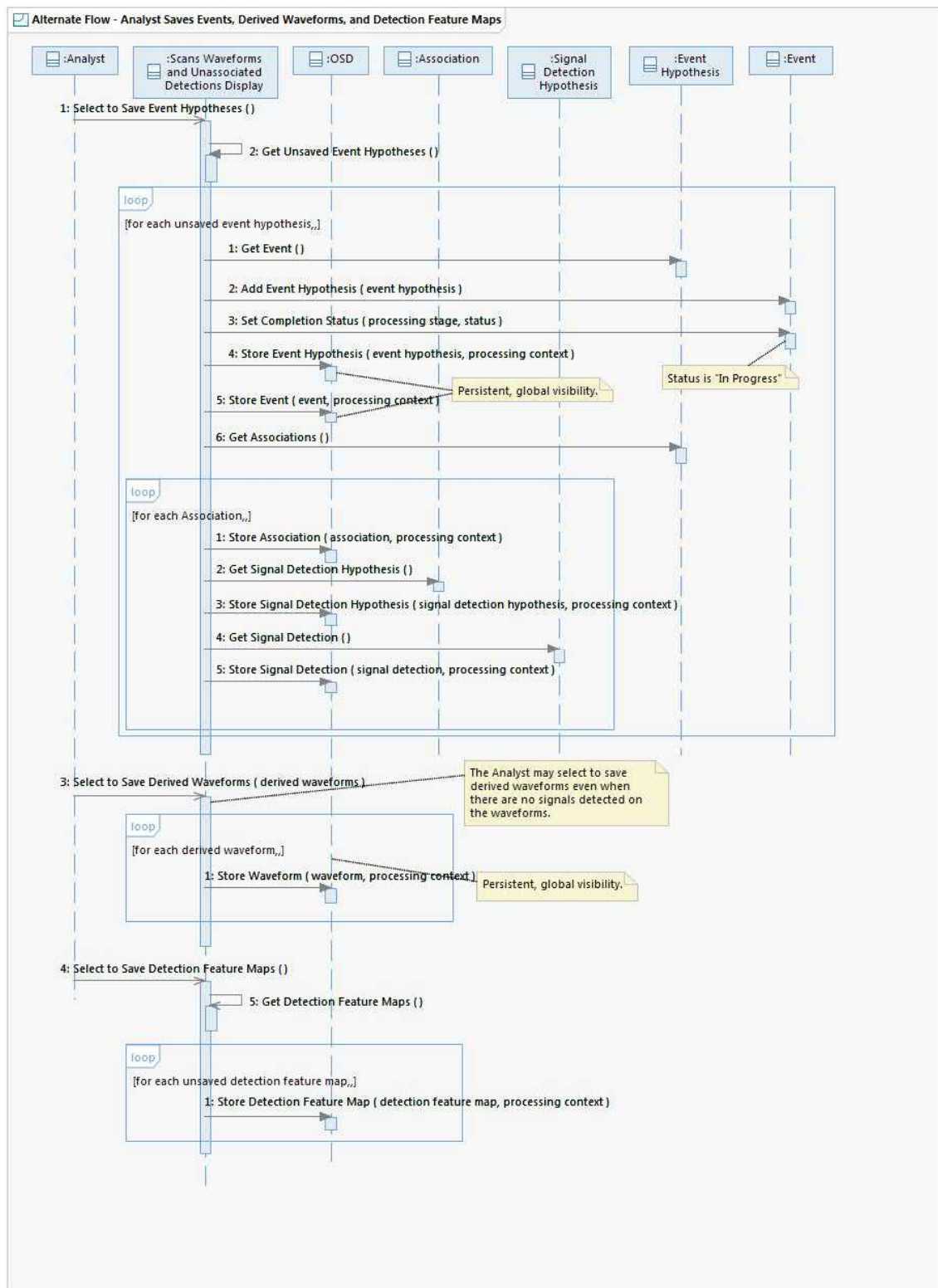
work that conflicts with, overlaps with, or otherwise affects the current Analyst's work. The Scans Waveforms and Unassociated Detections Display subscribes for Signal Detection, Waveform, Event, and Detection Feature Map callbacks to notify the Analyst of new or updated data. The Scans Waveforms and Unassociated Detections Display does not automatically update the display after notifying the Analyst of new or updated data. The Analyst may refresh the display to show the most recent data at any time (see "Alternate Flow – Analyst Refreshes Displayed Data").

### **Operation Descriptions**

None



## Alternate Flow - Analyst Saves Events, Derived Waveforms, and Detection Feature Maps





This flow shows the Scans Waveforms and Unassociated Detections Display using the OSD to save any Events, Event Hypotheses, and Detection Feature Maps that were created or modified by the Analyst. All of the Events are stored with an Event Completion Status of “In Progress”. See 'Refines Event' UCR and 'Marks Processing Stage Complete' UCR for details on how the Analyst can select a different Event Completion Status for an Event.

## **Operation Descriptions**

### ***Operation: Event::Set Completion Status()***

Set the Event Completion Status of an Event in the given processing stage to the given value.

### ***Operation: OSD::Store Event Hypothesis()***

Store the given Event Hypothesis with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

### ***Operation: OSD::Store Event()***

Store the given Event with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

### ***Operation: OSD::Store Waveform()***

Store the given Waveform with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

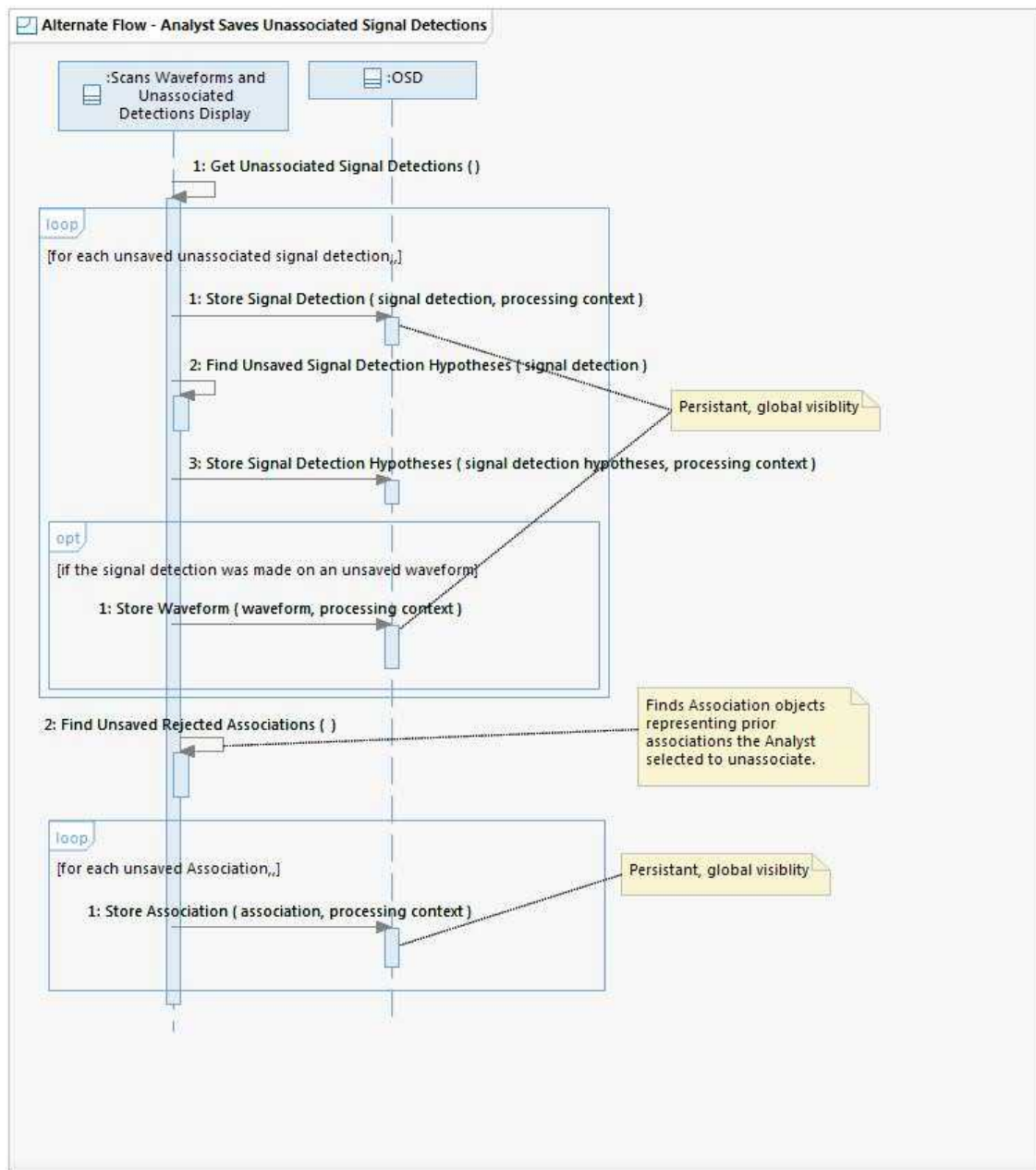
### ***Operation: OSD::Store Signal Detection()***

Store the given Signal Detection with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

### ***Operation: OSD::Store Detection Feature Map()***

Store the given Detection Feature Map with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

## Alternate Flow - Analyst Saves Unassociated Signal Detections



This flow shows the Scans Waveforms and Unassociated Detections Display using the OSD to save unassociated Signal Detections that were created or modified by the Analyst. The display also stores to the OSD any previously unsaved waveforms that contain saved signal detections.

### Operation Descriptions

#### **Operation: OSD::Store Signal Detection()**

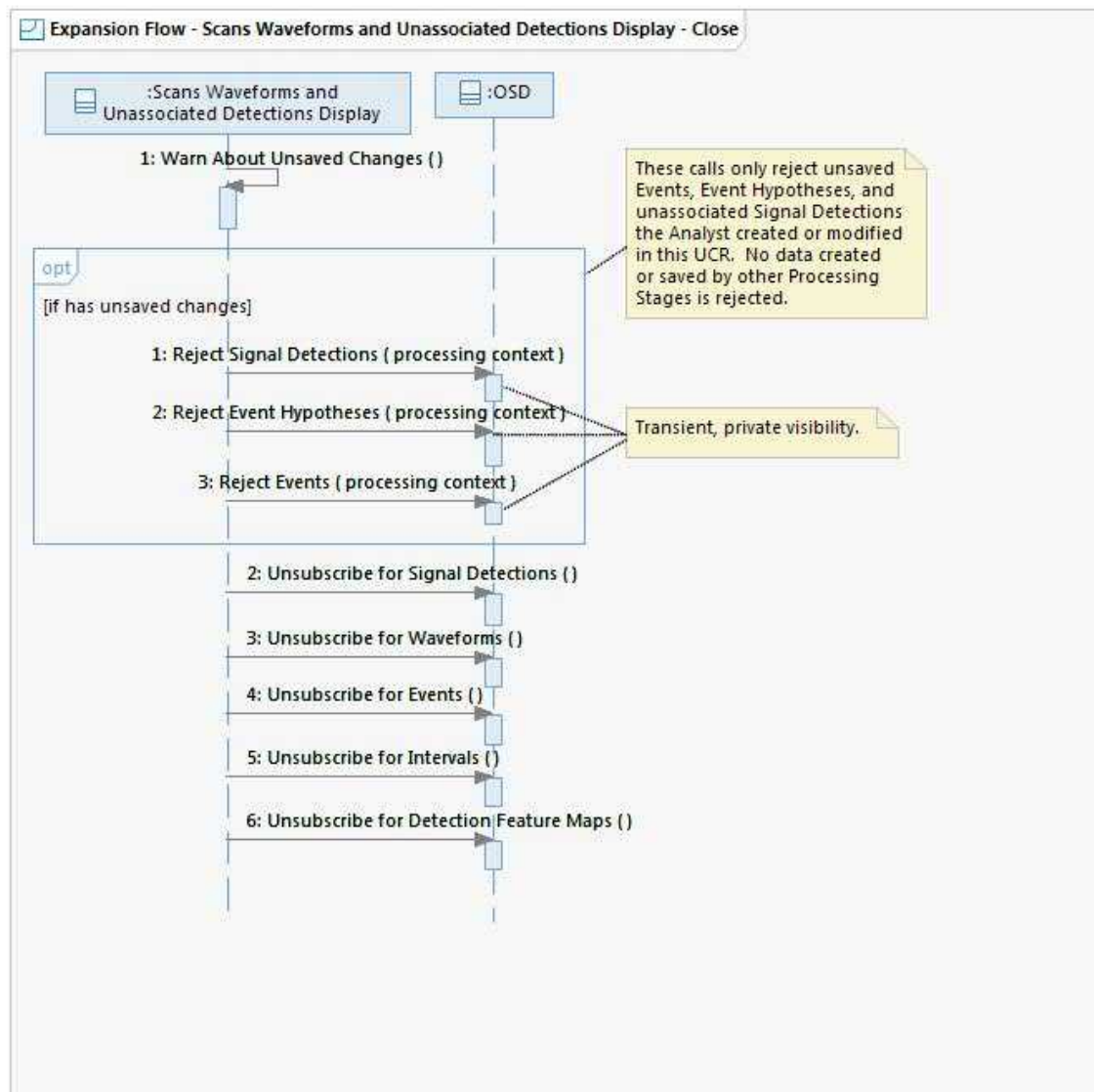
Store the given Signal Detection with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers

via callbacks.

**Operation: *OSD::Store Waveform()***

Store the given Waveform with the given lifespan (persistent vs. transient) and visibility (private vs. global) as specified by the given Processing Context and notify relevant subscribers via callbacks.

**Expansion Flow - Scans Waveforms and Unassociated Detections Display - Close**



This flow shows closing the Scans Waveforms and Unassociated Detection Display. The Scans Waveforms and Unassociated Detection Display checks for and warns the Analyst if there are any unsaved changes to Events, Event Hypotheses, or unassociated Signal Detections. The Scans Waveforms and Unassociated Detections display deletes the unsaved changes from the

OSD (“Alternate Flow - Analyst Saves Events and Unassociated Signal Detections” shows how the Analyst saves changes). The Scans Waveforms and Unassociated Detections display unsubscribes from OSD data updates.

## **Operation Descriptions**

None

## **State Machine Diagrams**

None

## **SSD Mappings**

### **General:**

**S-1157:** [*Objective / Priority 2*] The System shall provide the Analyst the capability to view newly acquired waveform data within 1 minute of acquisition.

**S-1284:** [*Threshold*] The System shall provide the Analyst the capability to view Waveform QC Masks.

**S-1372:** [*Threshold*] The System shall provide the Analyst the capability to view continuous beams.

**S-1373:** [*Threshold*] The System shall provide the Analyst the capability to view continuous beams for virtual event hypotheses for predefined geographic regions.

**S-1393:** [*Threshold*] The System shall store all derived channels related to one or more signal detections.

**S-1542:** [*Threshold*] The System shall not automatically perform network signal association affecting signal detections that the Analyst is actively reviewing.

**S-1859:** [*Threshold*] The System shall set the processing stage workflow status of a processing time interval to reflect analysis activity.

**S-1877:** [*Threshold*] The System shall notify Analysts working in a common processing stage if they are concurrently modifying signal detections in the same analysis time interval.

**S-1878:** [*Threshold*] The System shall provide the Analyst the capability to access and view all waveform data stored on the System.

**S-1885:** [*Threshold*] The System shall display 24 hours of continuous waveform data before the waveform displays flatline.

**S-1888:** [*Threshold*] The System shall provide the Analyst the capability to analyze continuous

waveform data from within a selected time block.

**S-1892:** [*Threshold*] The System shall provide the Analyst the capability to iteratively undo/redo back to the last saved state of an event.

**S-1893:** [*Threshold*] The System shall provide the Analyst the capability to sort channels based on distance from event.

**S-1894:** [*Threshold*] The System shall provide the Analyst the capability to sort channels based on station name.

**S-1895:** [*Threshold*] The System shall provide the Analyst the capability to sort channels based on station probability of detection for an event.

**S-1896:** [*Threshold*] The System shall provide the Analyst the capability to show all channels for a beam.

**S-1897:** [*Threshold*] The System shall provide the Analyst the capability to show all channels for an array.

**S-1898:** [*Threshold*] The System shall provide the Analyst the capability to show all channels for a 3-component station.

**S-1899:** [*Threshold*] The System shall provide the Analyst the capability to zoom the time axis of the waveform display.

**S-1900:** [*Threshold*] The System shall provide the Analyst the capability to scale the amplitude of individual displayed waveforms.

**S-1901:** [*Threshold*] The System shall provide the Analyst the capability to view a group of waveforms with their amplitudes scaled relative to the amplitudes of the other waveforms in the group.

**S-1902:** [*Threshold*] The System shall provide the Analyst the capability to adjust the height of an individual waveform display.

**S-1903:** [*Threshold*] The System shall provide the Analyst the capability to scroll waveform data along the time axis.

**S-1904:** [*Threshold*] The System shall provide the Analyst the capability to scroll the channels shown on the waveform display.

**S-1906:** [*Threshold*] The System shall provide the Analyst the capability to time align waveforms based on observed signal detections for a user selected phase.

- S-1907:** [*Threshold*] The System shall provide the Analyst the capability to time align waveforms based on real time.
- S-1908:** [*Threshold*] The System shall provide the Analyst the capability to align waveforms based on theoretical signal detections for a user selected phase.
- S-1915:** [*Threshold*] The System shall provide the Analyst the capability to process data without altering another Analyst's existing solution.
- S-1929:** [*Threshold*] The System shall provide the Analyst the capability to individually select signal detections as processing input during an analysis session.
- S-1947:** [*Threshold*] The System shall implement user interfaces according to the User Interface Guidelines.
- S-1986:** [*Threshold*] The System shall provide the System User the capability to view associated and unassociated signal detections on an interactive map.
- S-1987:** [*Threshold*] The System shall provide the System User the capability to view station data on an interactive map.
- S-2043:** [*Threshold*] The System shall store automatic and interactive processing results.
- S-2164:** [*Threshold*] The System shall access requested waveform data within one (1) minute of receipt by the Data Processing Partition.
- S-2166:** [*Threshold*] The System shall automatically process late-arriving waveform data within one (1) minute of receipt by the Data Processing Partition.
- S-2167:** [*Threshold*] The System shall write a 6 hour or less time block of 40Hz waveform data within the Operational Processing Time Period with a maximum 5 second latency. (Goal: 1 second.)
- S-2168:** [*Threshold*] The System shall read a 6 hour or less time block of 40Hz waveform data outside the Operational Processing Time Period with a maximum 10 second latency. (Goal: 2 seconds.)
- S-2169:** [*Threshold*] The System shall read a 6 hour or less time block of 40Hz waveform data within the Operational Processing Time Period with a maximum 5 second latency. (Goal: 1 second.)
- S-2170:** [*Threshold*] The System shall write a 6 hour or less time block of 40Hz waveform data from outside the Operational Processing Time Period with a maximum 10 second latency. (Goal: 2 seconds.)

**S-2223:** [*Threshold*] The System shall store all data and derived processing results to persistent storage as soon as the data and/or derived processing results are available.

**S-2420:** [*Threshold*] The System shall provide the Analyst the capability to store selected derived waveforms.

**S-2588:** [*Threshold*] The System shall set the processing stage workflow status of events to reflect analysis activity.

**S-2604:** [*Threshold*] The System shall provide the Analyst the capability to access late-arriving waveform data within one (1) minute of receipt by the Data Processing Partition.

**S-5610:** [*Threshold*] The Data Processing Partition shall access and process all waveform data stored on the system.

**S-5708:** [*Threshold*] The System shall read a 6 hour or less time block of processing results within the Operational Processing Time Period with a maximum 5 second latency. (Goal: 1 second.)

**S-5709:** [*Threshold*] The System shall write a 6 hour or less time block of processing results within the Operational Processing Time Period with a maximum 5 second latency. (Goal: 1 second.)

**S-5712:** [*Threshold*] The System shall read a 6 hour or less time block of processing results outside the Operational Processing Time Period with a maximum 10 second latency. (Goal: 2 seconds.)

**S-5713:** [*Threshold*] The System shall write a 6 hour or less time block of processing results from outside the Operational Processing Time Period with a maximum 10 second latency. (Goal: 2 seconds.)

**S-6437:** [*Threshold*] The System shall provide the Analyst the capability to time align detection feature maps based on signal detections.

**S-6438:** [*Threshold*] The System shall provide the Analyst the capability to time align detection feature maps with waveforms.

**S-6439:** [*Threshold*] The System shall provide the Analyst the capability to select signal detections on a detection feature map.

#### **IDC Specific:**

**S-5612:** [*Threshold*] The System shall provide the Analyst the capability to request auxiliary seismic waveform data from the Data Acquisition Partition.

## Notes

### General:

1. The Analyst refines Event Hypotheses created or modified by this UCR in 'Refines Event' UCR. When the Analyst creates or modifies an Event Hypothesis in this UCR, the System adds the Event to the list of Events requiring analysis (see 'Analyzes Events' UCR) with an Event Completion Status of "In Progress". The Analyst can then select to refine the Event from that list. See 'Marks Processing Stage Complete' UCR for a state machine diagram for Event Completion Status.
2. The Scans Waveforms and Unassociated Detections Display registers for OSD callbacks spanning the timeframe of a Processing Activity Interval. These callbacks do not include buffers around the Processing Activity Interval's timeframe.



# **IDC Use Case Realization Report**

## **UCR-14.05 Performs Expert Technical Analysis**

### **Use Case Description**

This architecturally significant use case describes how the Fusion and Review Officer conducts an expert technical analysis upon request from a member state. The request could be for expert technical analysis for either a more detailed analysis of a specific event using IMS data and software (see 'Analyzes Event' UC) or for specified analysis using member state specified data and/or software (see 'Analyzes Research Event' UC).

The Fusion and Review officer creates an Updated Event Bulletin when doing a more detailed analysis of the IMS data using System tools. The Fusion and Review Officer creates a State Requested Methods Report (SRMR) that is based on the results of the extensive analysis using the member state specified data and/or software. Both types of analysis may include but are not limited to geographic location, time, depth, magnitude, waveforms, the signal detection list, a map, phase and depth refinement, and focal mechanism. The Fusion and Review Officer makes the Updated Event bulletin, SRMR and member-state provided data and software available to all member states (see 'Views System Results' UC).

This use case is architecturally significant due to the inclusion of data and software provided by member states.

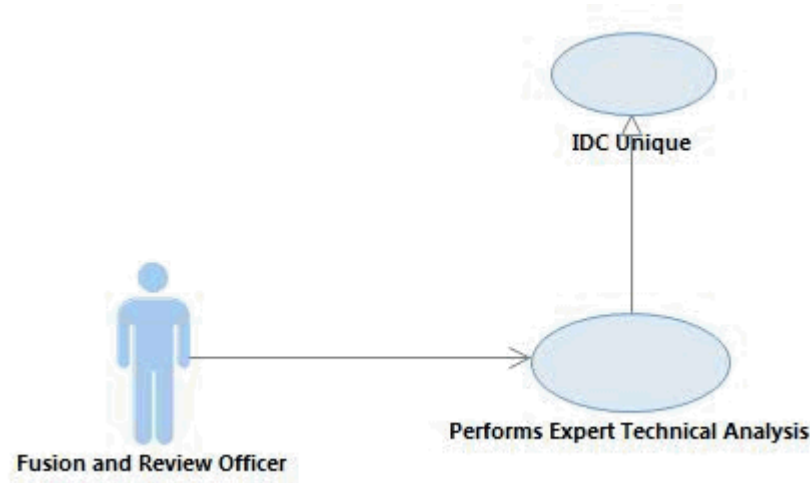
### **Architecture Description**

The Fusion and Review Officer (FARO) performs expert technical analysis by invoking either the Analyzes Events UCR or the Analyzes Research Events UCR. When a member-state requests expert technical analysis using additional data or software, the FARO uses Analyzes Research Events UCR; otherwise, the FARO does the requested extra analysis via Analyzes Events UCR.

Once the FARO completes the requested analysis, the FARO opens the State Requested Methods Report (SRMR) Display to create the SRMR. Using the SRMR Display, the FARO creates the SRMR, updates its contents with the results of the expert technical analysis and releases the SRMR.

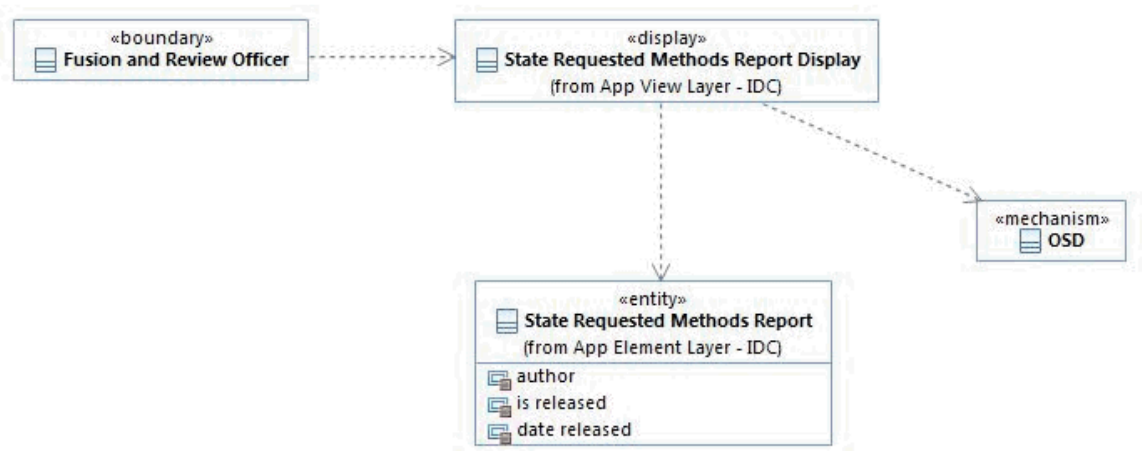
The FARO finds and views previously created States Requested Method Reports via the SRMR Display.

## Use Case Diagram



## Class Diagrams

### Classes - State Requested Methods Report Display



This diagram depicts the State Requested Methods Report Display and related classes. The Fusion and Review Officer opens this display to make and view State Requested Methods Reports, which the display stores and retrieves via the OSD.

## Class Descriptions

<<boundary>> *Fusion and Review Officer*

The Fusion and Review Officer actor is an Analyst who performs expert technical analysis.

***<<display>> State Requested Methods Report Display***

Display for creating, editing, and viewing State Requested Methods Reports.

## <<entity>> State Requested Methods Report

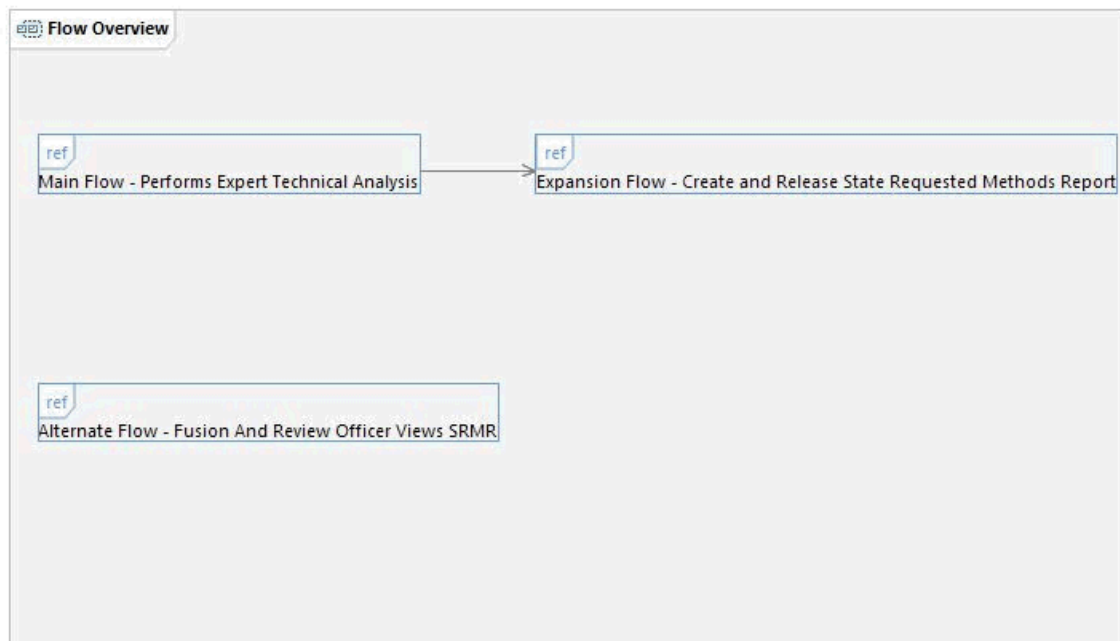
Entity representing a report of analysis undertaken by a Fusion and Review Officer per request from a member-state, possibly using extra data and/or software provided by the member-state.

**<<mechanism>> OSD**

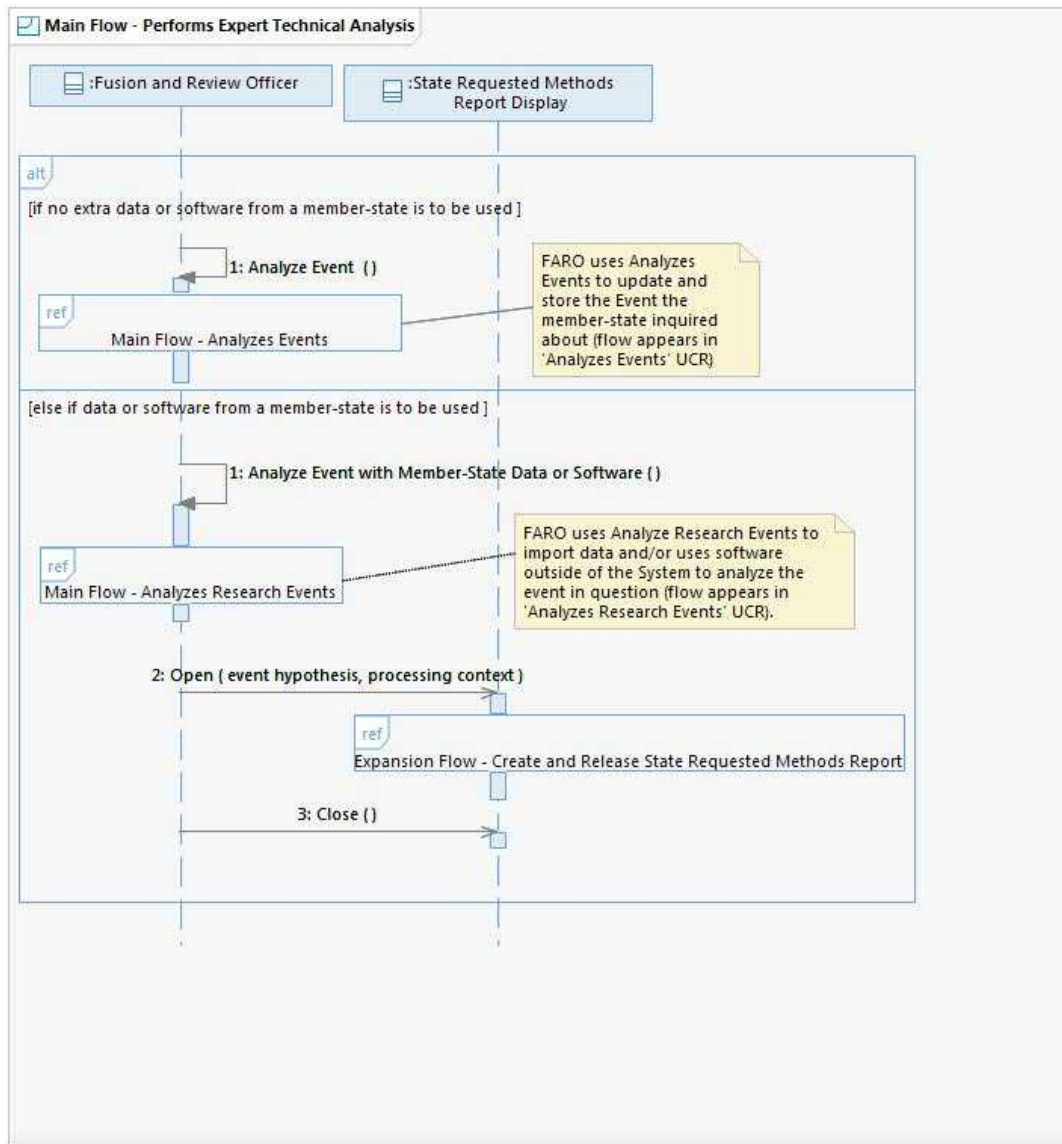
Represents the Object Storage and Distribution mechanism for storing and distributing data objects internally within the system.

## Sequence Diagrams

### *Flow Overview*



## *Main Flow - Performs Expert Technical Analysis*



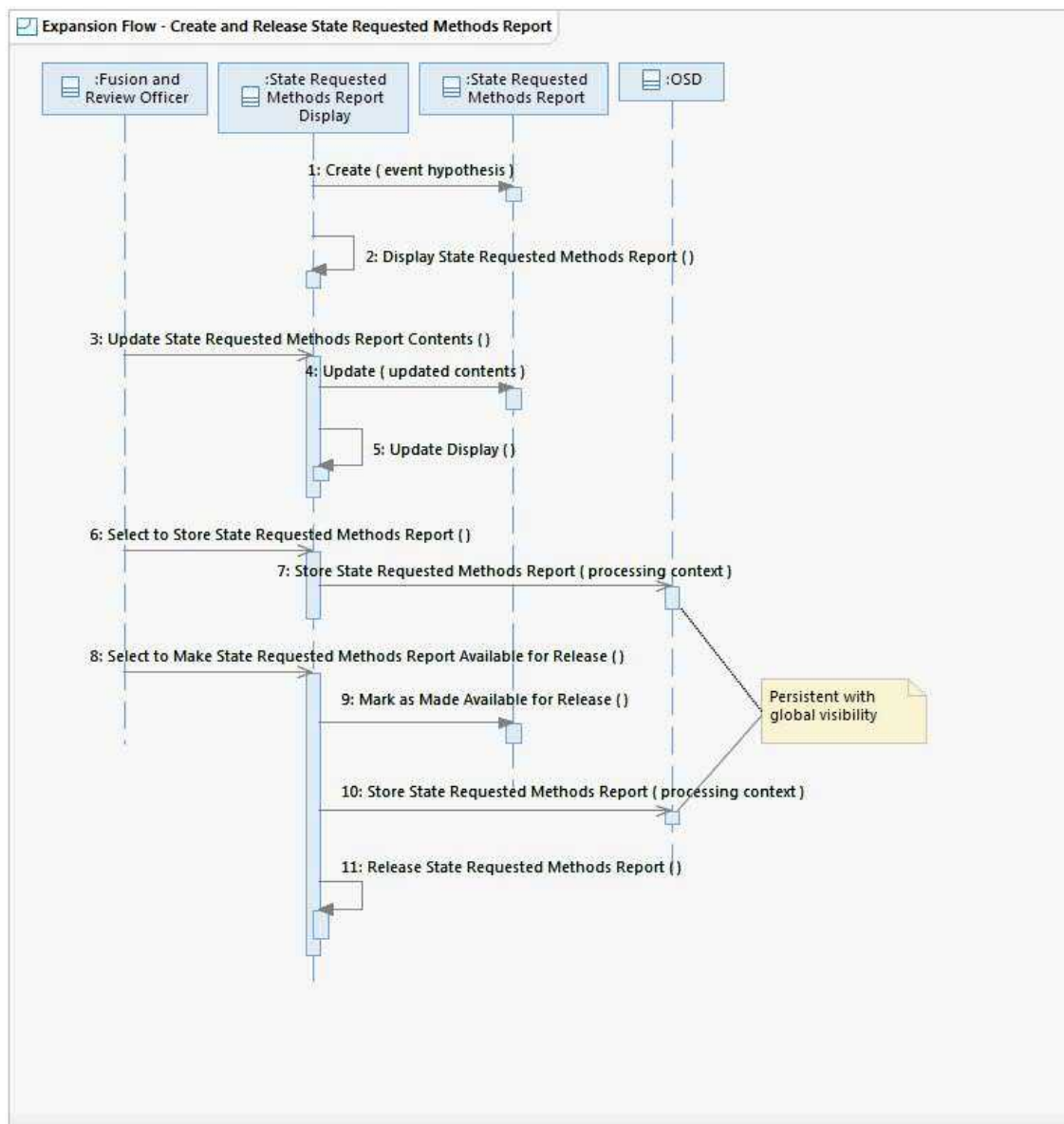
This flow illustrates how the Fusion and Review Officer (FARO) goes about performing the expert technical analysis of a particular event at the behest of a member-state. The FARO invokes either the Analyzes Event UC or Analyzes Research Events UC in order to refine the event; the Analyzes Research Events UC is used when the member-state requests that additional software or data be included in the analysis by the FARO. After performing the analysis on the event and saving it, the FARO opens the State Requested Methods Report Display to create the State Requested Methods Report (SRMR), update it with the results of the analysis as the FARO sees fit, save it to the OSD, and then selects to make the SRMR available for release.

## Operation Descriptions

### ***Operation: State Requested Methods Report Display::Open()***

Opens the display. This display may be opened from a System display (e.g. Refines Event Display), or as a separate application.

### ***Expansion Flow - Create and Release State Requested Methods Report***



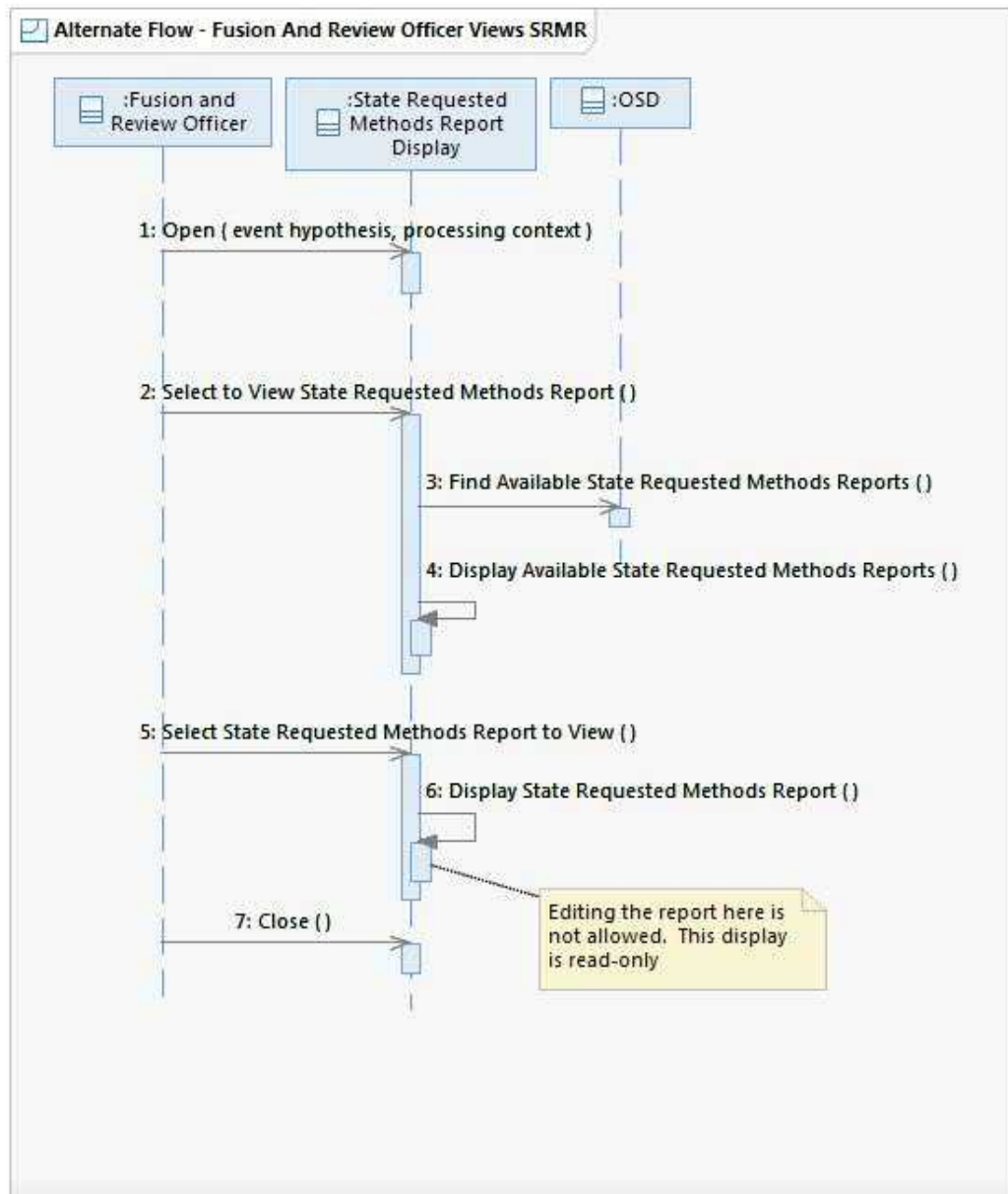
This flow shows how the Fusion and Review Officer interacts with the State Requested Methods Report Display to create, edit, store, and release a State Requested Methods Report.

## Operation Descriptions

### ***Operation: State Requested Methods Report Display::Update State Requested Methods Report Contents()***

Add contents to the State Requested Methods Report, such as text and images that describe the results of the technical analysis of the event.

### ***Alternate Flow - Fusion And Review Officer Views SRMR***



This alternate flow shows how the Fusion and Review Officer can find and view State Requested Methods Reports through the State Requested Methods Reports Display.

## Operation Descriptions

### ***Operation: State Requested Methods Report Display::Open()***

Opens the display. This display may be opened from a System display (e.g. Refines Event Display), or as a separate application.

## State Machine Diagrams

None

## SSD Mappings

### **General:**

**S-1947:** [*Threshold*] The System shall implement user interfaces according to the User Interface Guidelines.

### **IDC Specific:**

**S-6457:** [*Threshold*] The System shall provide the Fusion and Review Officer the capability to create an event report.

**S-6458:** [*Threshold*] The System shall provide the Fusion and Review Officer the capability to create a State Requested Methods Report.

**S-6459:** [*Threshold*] The System shall provide the Fusion and Review Officer the capability to save an event to the Updated Event Bulletin.

**S-6460:** [*Threshold*] The System shall provide the Fusion and Review Officer the capability to distribute a State Requested Methods Report.

## Notes

### **General:**

None

### **IDC Specific:**

1. The Updated Event Bulletin and State Requested Methods Report (SRMR) are processing results that are available for access by an Authorized External User (see 'Views System Results' UC and 'Requests System Data' UC).

